Compensation Techniques for Network Mismatch in Telephone-Based Speaker Verification

Ulrich Türk

Lehrstuhl für Mensch-Maschine-Kommunikation
Technische Universität München

Compensation Techniques for Network Mismatch in Telephone-Based Speaker Verification

Ulrich Türk

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr. sc. techn. (ETH) Andreas Herkersdorf

Prüfer der Dissertation:

1. apl. Prof. Dr.-Ing., Dr.-Ing. habil. Günther Ruske
2. Univ.-Prof. Dr.-Ing. Georg Färber

Die Dissertation wurde am 24.10.2007 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 22.01.2008 angenommen.

Zusammenfassung

In dieser Arbeit wird ein neuer Ansatz für die Adaptierung von automatischen Sprecher-Verifikations-Systemen entwickelt, der ihre Robustheit gegenüber dem Wechsel von Telephon-Sprache aus dem Festnetz und aus dem GSM-Netz erhöhen soll. Bei den meisten sprachverarbeitenden Erkennungssystemen wirken sich Schwankungen der akustischen Eigenschaften des Sprachsignals negativ auf die Erkennungsleistung aus. Der Effekt tritt besonders dann hervor, wenn zum Erkenner-Training und zur Performanz-Messung Sprachdaten aus unterschiedlichen akustischen Umgebungen verwendet werden. Konkret befassen wir uns in dieser Arbeit mit den Sprecher-Modellen, die mit Daten von Festnetz-Telephonen trainiert wurden und die zur Sprecher-Verifikation mit Sprachaufnahmen aus dem GSM-Netz eingesetzt werden sollen. Aus der Vielzahl von Modellierungstechniken haben wir Hidden Markov Modelle gewählt.
Der neue Ansatz in unserem Verfahren ist, die Original-Sprachdaten durch den GSM-Codec zu kodieren und wiederum zu dekodieren, um somit einen Satz von simulierten GSM-Sprachdaten zu erhalten. Zusammen mit den originalen Sprachaufnahmen lassen sich auf Basis des MLLR-Algorithmus Parameter-Sätze berechnen, die die Sprechermodelle an die neue akustische Umgebung adaptieren. Die Adaption kann sowohl für das gesamte Sprechermodell mit einem einzigen Parameter-Satz erfolgen, als auch mit mehreren Parameter-Sätzen, die auf phonetisch oder akustisch bestimmte Untergruppen im Modell wirken. Unser Ansatz ist speziell für die Adaptierung in Sprecher-Verifikations-Systemen geeignet, da hier oft nur eine kleine Anzahl von Sprachaufnahmen für das Training und die Adaptierung zur Verfügung stehen.
Im ersten Teil unserer Arbeit gehen wir kurz auf die Performanz-Parameter von Sprecher-Verifikations-Systemen ein und zeigen verschiedene Ansätze, um die Konfidenz-Intervalle dieser Parameter zu bestimmen. Da oftmals die Unterschiede zwischen experimentellen Meßwerte in der Größenordnung der Varianz dieser Parameter liegen, konzentrieren wir uns daher zuerst auf die Auswahl eines geeigneten Verfahrens zur Schätzung der Konfidenz-Intervalle.
Im zweiten Teil untersuchen wir zunächst die Effekte des GSM-Codecs im Merkmalsraum von kepstralen Koeffizienten. Wir zeigen die Motivation für die Verwendung von mehreren Parameter-Sätzen in unserem Adaptions-Verfahren, vergleichen unseren Ansatz mit etablierten Verfahren aus der Literatur und gehen auf die mathematische Formulierung des Verfahrens ein.
Abschließend evaluieren wir unser Verfahren mit Hilfe eines speziellen Korpus für Sprecher-Verifikation mit deutscher Sprache, dem VeriDat-Korpus.
Unser Ansatz zeigt eine größere Verbesserung der Verifikations-Leistung des Erkenners als ein Standard-Verfahren, das die Sprechermodelle mit dem Baum-Welch-Algorithmus neu trainiert. Es zeigt sich jedoch, daß unser Verfahren nur bei Verwendung von sprecher-spezifischen Parameter-Sätzen erfolgreich ist. Die Adaptierung kann nicht die Performanzwerte erreichen, die durch das Training der Sprechermodelle mit gemischten Daten möglich sind.

Abstract

In this work a novel approach is presented for adapting an automatic speaker verification system to the mismatch between speech from fixed telephone channels and GSM channels. Typically the performance of speech related recognition tasks degrades when the acoustical characteristics of the training set differ from those of the evaluation set. Our work concentrates on the case where the speaker models are trained on fixed telephone data and the verification takes place with recordings from the GSM network. In addition we focus on models based on Hidden Markov Models.
The novel idea of this approach is that the original speech data is fed through the GSM coding and decoding stages in order to derive simulated GSM speech. Together with the original speech, transformation parameters based on MLLR are calculated that allow the adaptation of the speaker models to the new acoustical environment. The adaptation can be performed with a single parameter set affecting the complete speaker model or by several sets which operate on acoustic or phonetic sub-clusters of the model. Our approach is especially designed for adaptation for the typical situation in speaker verification systems where only a small amount of speech data for model training and adaptation is available.
In the first part of our work we present shortly the performance parameters of a speaker verification systems and several approaches for estimating their confidence bounds. Since the measured performance differences are typically of the size of the statistical confidence intervals we find it crucial to evaluate different techniques for estimating the confidence bounds.
The second part starts with an investigation of the effects of the GSM codec in the domain of the cepstral features. We explain the motivation for the individual adaptation of sub-clusters in the model, compare our approach with related adaptation techniques from literature and present the mathematical foundations of our adaptation technique.
Finally we evaluate our approach using the German VeriDat corpus for speaker verification. We find that our proposed adaptation technique leads to a higher performance improvement than the standard Baum-Welch retraining technique. However, the adaptation is only successful when calculating parameter sets for each individual speaker. In addition, the adapted models can not reach the performance of models trained on mixed speech data.

Contents

Chapter 1

Introduction

Within the past 15 years, our society became more and more networked. People access increasingly local and global communication and information services instead of personally addressing to local offices or agencies like e.g. banking institutions or civil services. Since many of these services require a proper identification or authentication of the client, reliable techniques have to be developed in order to avoid the security threat of impersonation. These techniques are subsumed under the term *verification* which describes the service of verifying a claimed identity of an individual. Traditional verification techniques like e.g. comparing visually the photograph of an identity card with the present person are error-prone and hardly feasible for remote services. This is the point where biometric technologies become applicable since they deploy either physiological or behavioural characteristics of a person that can be easily captured and compared to stored templates of the same person.

Among various biometric properties, the human voice can be attributed as the most natural biometric feature: it is besides visual clues the primary source for humans to identify individuals. First research in this area was mainly motivated by forensic applications. In 1962, Kersta [1] introduced the term *voiceprint* when he investigated the usage of spectrograms as a mean of personal identification. One year later, Pruzansky [2] was among the first to compute the distance between two digitally spectrograms in order to determine automatically the speaker's identity.

Over the time, many approaches for automatic speaker verification and identification (or short speaker recognition) systems have been developed. While the first attempts deployed long-term features and statistical parameters of the speech, later developments in the 1970s started to use frame-based measures like e.g. linear predictive coefficients and cepstral coefficients as features. In the 1980s, modelling with vector quantisation (VQ) and Hidden Markov Models (HMMs) replaced the template matching methods. Since the 1990s, further improvements have been introduced such as text-independent methods using Gaussian mixture models (GMMs), score normalisation methods, robustness to utterance variations and inclusion of high-level features (e.g. pronunciation or phone usage).

Speaker recognition is inherently the ideal biometric feature for any telephone-based service. Not only forensic applications exist but also various commercial fields like e.g. access to all kinds of personalised services (bank account, civil services, etc.). This is the reason that many publications deal with the bandlimited speech of the telephone data and several speech corpora with telephone data has been created for this special purpose.

Research in speaker verification and identification has always been related closely to research in other speech technologies, mainly speech recognition. Today, one of the main topics in these fields is still the environmental robustness. While robustness to utterance variations can now be

easily achieved with clean and well-matched speech, other acoustic effects remain as a challenge: clients of a speaker recognition systems tend to use different handsets (including cellular phones) in different situations i.e. with varying acoustic background noises. The recognition performances drops notably when different conditions are used for enrolment and application which makes speaker recognition over telephone networks a large research topic.

Whenever sufficient speech data is available from an unseen acoustic environment, it is still the best solution to retrain the involved models i.e. the speaker model and optional background models used for normalisation purposes. Often a multi-style training [3] is used in order to incorporate all possible environmental effects into the models. However, fine details in the speaker properties will be blurred by pooling data from different environments.

Since collecting data from different environments means also increased effort for the client speaker, different adaptation methods have been investigated for speaker recognition. Standard adaptation techniques like e.g. MAP (maximum a posteriori) training or MLLR (maximum likelihood linear regression) allow to transform existing models and achieve often a notable performance gain with less data compared to the retraining approach.

We will later see that many publications have already dealt with the adaptation of speaker models when different transducers are used for the recordings. However, to our knowledge the mismatch between fixed line telephone data and GSM telephone data together with the investigation of adaptation methods for its compensation has not been addressed in detail for speaker verification systems.

An important initiation for our work has also been the fact, that the Institute of Phonetics and Speech Communication, University of Munich, was involved in the validation of the VeriDat database, which gave us access to this outstanding corpus for speaker verification. Since it includes telephone recordings of 150 speakers from both networks, it served us as an ideal starting basis for our investigations.

Overview

Our presentation comprises two parts: in the first part, including chapter 2 to 6, we present the foundations of speaker verification systems and their performance evaluation. We continue with the results of preliminary experiments on the robustness of different speech features. These results serve as primary motivation for the further work that we lay out in the second part (chapter 7 to 10). Here we propose an adaptation technique for speaker models, compare it with related work from literature and discuss its performance under varying setups.

We start with chapter 2 where we give a short definition of biometrics, its applications and the different biological properties that are deployed for biometrics. We focus then on the voice as a biometric feature and present finally the inherent errors that occur in verification systems and the typical groups among the users that contribute with varying degree to these errors.

In chapter 3 we outline different classifications of speaker verification systems depending on their application (verification versus identification) or on the linguistic content (text-independent, text-dependent or text-prompted). Since we focus in our work on the verification process we present the basic setup of a speaker verification system that is shared by many systems in research and applications. For commonly deployed processing stages we depict the mainly applied techniques: for the feature extraction stage, we present the fundamentals of the cepstral features, for the speaker modelling, we focus on models based on the Hidden Markov Models (HMMs for text-dependent SV systems and GMMs for text-independent SV systems) and for the feature comparison we present different score normalisation techniques. For the final decision taking stage we give references to additional score normalisation approaches such as Z-norm, H-norm and T-norm. In the last section we outline our MASV system, an experimental SV system

based on HTK (Hidden Markov Model Toolkit) [4] that was created during our work in order to conduct various experiments.

Since the understanding of performance measures is crucial for comparing results of different experiments, we present in chapter 4 commonly used performance measures. First we refer to sources that have a general impact on the performance of any speaker verification system. Then we introduce the well known measures 'false rejection rate' (FRR), 'false acceptance rate' (FAR), 'equal error rate' (EER) and the DET plots. We continue with different concepts that capture the inherent statistical variation of these measures including the binomial approach, the McNemar test, and a sub-sampling approach proposed as best practice by the National Physical Laboratory [5]. We give a short introduction to the bootstrap principle, that allows to determine empirical confidence bounds of the performance measures and present practical aspects for the construction of confidence bounds for the EER and the DET plots. The last section deals with training and evaluation schemes that determine the speakers and the sessions that are used during the training of the system and for its evaluation.

In chapter 5 we introduce an extended scheme for estimating the confidence bounds of the FAR which is based on the random effects model. We will then compare the different approaches to predict the confidence bounds for the FRR and FAR with empirical bounds from bootstrap experiments. The basic variability models for our performance measures, the binomial approach and the McNemar test, turn out to be far too optimistic about the real confidence bounds that we find from the bootstrap experiments. Similar to other work from literature we find a speaker effect for the genuine speaker tests (relevant for the FRR) and effects for both the impostor speaker and the claimed speaker model for the impostor speaker tests (relevant for the FAR). Therefore our sub-sampling approach for modelling the variability of the FRR, FAR and the DET plot seems to be appropriate. It will be used later on in chapter 10 in order to compare the statistical relevance of performance differences.

Chapter 6 presents preliminary experiments on the robustness of the two main types of cepstral features, the MFCCs (mel frequency cepstral coefficients) and the LPCCs (linear prediction cepstral coefficients). Since the VeriDat corpus provides recordings from varying telephone networks (fixed line versus GSM) and two levels of background noises (quiet versus noisy), we can measure the robustness with regard to both domains separately. We introduce a robustness coefficient which allows us to judge the robustness easily for different setups and compare it to related work from literature. Finally we find in our experiments that the LPCCs are already sufficiently robust against variations in the level of the background noise but still suffer from mismatch in the network domain. In addition, our preliminary experiments show that a simple approach by training separate speaker models for each network domain does not lead to a performance improvement.

For the second part of our work, our goal is to develop an adaptation scheme for a LPCC-based SV system to cope with a mismatch in the network domain. We start in chapter 7 with a brief presentation of the basic properties of the GSM codec and its effects on the speech signal: we simulate the coding and decoding of the GSM standard and use the transcoded speech to investigate the differences between the fixed line speech and the transcoded speech. We calculate the Bhattacharyya distance to quantify the effects that we find for different phonetic segments.

In chapter 8 we summarise shortly different adaptation and transformation techniques that are known from literature, among them MAP (maximum a posteriori) training, MLLR and feature-based transformations. Based on our findings from the previous chapter we present our own adaptation scheme, which is based on the MLLR approach and deploys simulated GSM data for the adaptation process. Since MLLR allows the use of regression classes, we can easily compensate phoneme-dependent effects of the GSM codec.

Finally in chapter 9 we give detailed experimental results of our adaptation scheme with varying parameters. We investigate wether the basic concept works by using a simplified variant of the adaptation which uses simulated GSM speech both for adaptation and evaluation of the SV system. Then we will present the dependency of the adaptation performance on the complexity of the transformations and the utility of MLLR regression classes and speaker-independent transformation sets. Finally we investigate which effects in the real GSM speech data can be attributed to the GSM speech codec and if the remaining effects are speaker dependent or could be compensated by a common transformation set calculated from real GSM speech.

Chapter 10 compares selected results from our adaptation experiments with simple retraining methods such as retraining with simulated GSM data or training with a mixed data set including fixed line recordings and real GSM recordings.

In chapter 11 we will provide a summary of our works and give an outlook on potential further work in this area.

The appendix includes detailed information about the VeriDat database and its speaker population, the session lists for the various training and evaluation setups, the HTK settings for the speech features, parameter lists used for the GSM codec simulation, comprehensive tables with results of the adaptation experiments and finally an overview of the formulary.

Chapter 2

Biometrics

This chapter gives a brief introduction to the main features of biometrics and commonly deployed biometric technologies. *Biometrics* usually denotes automatic recognition technologies for identifying and verifying persons based on data given by the human nature. We will outline currently deployed data sources and focus then on the human voice as biometric feature. Finally we will present briefly the main performance measures and point out typical performance differences within the user population of a biometric system.

A good introduction into biometrics can be found e.g. in [6, 7].

2.1 General approach to biometrics

Among the most dangerous security threats in our networked society is impersonation i.e. somebody claims to be someone else. Many services require reliable identification and authentication of principals before access can be granted to locations or services.

Both tasks, identification and authentication, are closely related but differ in the type of application and the technological constraints. *Identification*, also often denoted as *detection*, is the task to analyse a biometric sample and to assign it to a person from a known set of individuals. Its is typically applied in forensic cases or when a technical system should react in a personalised way and security issues are not relevant.

In contrast, *authentication* services verify a claimed identity by comparing the biometric sample with previously collected knowledge about an individual. The term authentication is often used interchangeable with the term *verification*. Generally, biometric protection of a system against unauthorised use is performed by authentication. Protection comprises access control to location, facilities or also to data and data services. Three main approaches for authentication procedures are deployed:

- Proof of knowledge. The verifier knows information about the claimed identity that can only be known by a principal with that identity. Passwords, PINs or questionnaires fall under this category.
- Proof of possession. The principal is authorised by possession of an object e.g. smart card, RF-ID batch or a key.
- Proof of property. The principal provides certain properties of human characteristics that are measured. Techniques for measuring and judging these properties are subsumed under the term *biometrics*.

Authentication by the first two approaches is commonly used in many facilities e.g. banks, industry or government. However, techniques based on these authentication procedures are the more unreliable ones as objects can be stolen and knowledge can be spyed. Questionnaires can be very weak when someone knows the user well enough. Finally many objects for proof of possession can be copied e.g. magnetic stripe cards. In combination with a stolen PIN someone can impersonate as somebody else easily.

Especially in the financial sector fraud is increasing rapidly e.g. credit card fraud. Skinning of automatic teller and cash machines becomes a new fraud scheme, where data like the PIN or information on the card are grabbed by a false user interface installed on top the real one.

Biological properties deployed in biometrics

Biometrics denotes the statistical exploitation of data provided by the human biology. The personal identity is thus defined by its individual biological or behavioural characteristics. The advantage is that some human biological properties are unique and are hardly separable from a human being.

Commonly used biometric features are:

- Fingerprint analysis
  It is regarded as the oldest application of biometrics. In 1870, Alphouse Bertillon (France) invented a system for identifying criminals based on finger print analysis. The so-called Bertillon system uses a classification based on certain characteristics (arch, loop, whorl) of the ridges of a fingerprint. The patterns are different for each individual and remains generally unchanged during lifetime.

  Fingerprint systems are used in law enforcement and control in welfare programs, especially in the USA. Fingerprint verification is associated by users with criminality and thus public acceptance is low. In order to avoid this association, templates should be stored on user owned media e.g. smart cards and not in central databases.

  The devices need special scanner types that should be able to detect replicates of fingerprints or to perform live tests in order to achieve high security levels.

  Application of this technique in some working environments, e.g. where workers wear gloves (medical or chemistry laboratories), will not be appropriate.

- Iris analysis
  Clinical results showed that the pattern of every iris is unique and it remains unchanged during live time. The iris consists of a meshwork of connective tissue, collageneous fibres, furrows, rings and coloration. All these constitute a distinctive pattern that can be used for identification or verification. The properties of the iris are especially suitable as it is well protected from the environment, it can be registered without physical contact and its physiological response to light provides a natural test for manipulation. Systems based on iris analysis achieve decision about individual identity with high statistical confidence.

  However, public acceptance of the iris analysis is rather low since the small distance of the capture system to the eye is felt as more intrusive compared to other biometric techniques.

- Facial analysis. It is together with voice verification the most natural recognition technique as it is also the most traditional way of personal identification. Face characteristics e.g. size of the nose, eyes and mouth and their relative position to each other are unique for an individual. These features are extracted from camera pictures which can be either snap-shots

or live video. The public acceptance is claimed to be quite high. However this technique has difficulties with unexpectedly imposed characteristics such as beards or glasses.

- Hand geometry. Distinct characteristics of the hands including the external contour, internal lines and blood vessel patterns on the back of the hand can be used to distinguish individuals. Again the performance of these systems will be affected by disturbances e.g. rings or swollen fingers. This biometric technique is acceptable in most countries; however, it was found that e.g. people in Japan do not like to place their palm where other people do.
- Signature verification. Signing is a well trained action which is not influenced by deliberate muscle control. The characteristics of this act e.g. rhythm, velocity and acceleration form an individual pattern which is hard to reproduce by a different person. Input media can be either specialised pens or tablet based systems using special surfaces to collect the data. Signature analysis is highly accepted as it is a familiar way to authenticate a document or a person. Naturally it has difficulties with people changing their signature frequently. Application in countries with high illiteracy rate is not possible.
- Voice verification. Humans use the characteristics of the voice naturally to identify someone. The various properties of speech sounds and also non-speech sounds are formed by articulation and physiological features. We will elaborate the voice verification in section 2.2.

Relatively new biometric methods comprise DNA pattern analysis, ear recognition, key stroke analysis and head geometry analysis.

Categorising biometric features

Biometric techniques can be categorised into two classes: physiological based techniques and behavioural based techniques. The former ones are based on physiological properties of a person, that are often very stable over time and assumed to be unique for an individual person. The latter ones measure the behaviour of a person who is performing a certain task e.g. typing on a computer keyboard or speaking. Due to the influence of the state of health, emotional state and drift over time, these features suffer from a greater variability compared to the physiological features.

Systems based on physiological techniques, especially fingerprint and retinal analysis, are more accurate however the devices are larger and more expensive. Despite their high accuracy they are perceived by the public as less acceptable for daily usage. Behaviour based systems are embedded more unobtrusively in a verification process and are thus more easily accepted by the user. However, these systems need more refined techniques to handle individual variability of the features. The required sensors are often cheaper compared to those of the physiological systems but more sophisticated techniques are necessary for tackling the high feature variability.

General requirements to biometric features

Like any pattern recognition system, biometric techniques use a feature extraction stage that serves as input for a classifier. The goal of the feature extractor is to characterise objects by measurements that allow the classifier to group similar objects together while separating different objects from each other. Both stages are highly depending on each other: virtually perfect features would render the classifier rather trivial while an omnipotent classifier would cope also with weakly distinguishing features. Depending on the task on the nature of the data

source, ideal features can not be found in most cases. The general requirements for biometric features would comprise:

- universal property that can be found in all members in the contributing community
- uniqueness, depending on the application even for identical twins
- quantitatively measurable
- robust to short-term and long-term variability of the measured property, leading to compact clusters for an individual
- low dimensionality for low-complexity classifiers
- robust to variability of the environment during measurement
- robust to impersonation

2.2 Voice as biometric feature

The human voice is together with the facial image the most common biometric characteristic used by humans to identify other persons. Voice recognition fits into both types of biometrics defined above: behavioural and physiological biometrics. On the one side, the human voice does not exist without the effort of producing speech i.e. showing one's speaking behaviour. On the other side, various characteristics of the voice are formed by physiological factors of the vocal system.

Speech is a medium that not only communicates a messages (what is said) but also transmits additional information about the speaker (who said it) and the speaker's attitude (how it is said). Although humans can separate these three types of information easily, it is far from trivial for a pattern recognition system to extract only the speaker information.

The speaker characteristics are intermingled on the acoustical domain of speech with the linguistic content and the speaking style due to different attitude, emotional state or state of health. These obscuring components with regard to the speaker characteristics are accounted to the intra-speaker variations.

The linguistic content varies due to the context of the dialogue or dialects and regional variants. Although speaker recognition can be performed on a linguistic level as e.g. some speakers use specific words or specific phrases, these clues are not very stable as the speaker might rephrase his utterances next time.

Various voice characteristics are determined to a great part by the size and shape of the vocal tract. Therefore commonly used features like e.g. filterbank and cepstral coefficients capture the shape of the speech spectrum. We will present the cepstral features, which were deployed in our experiments, in more detail in section 3.2.

2.3 Performance of biometric systems

The key property for judging the performance of a biometric system is its error rate. For verification systems, two types of errors exist: the system might reject a claimed identity that should have been accepted (false rejection of a client) or accept a claimed identity that should

have been rejected (false acceptance of an impostor). Most biometric systems deploy a variable decision threshold that controls the tradeoff between false rejections and false acceptances. Both types of errors are presented in more detail in chapter 4.

Generally when judging these two errors, the conditions of the application scenario of the biometric system must be taken into account. In contrast to forensic applications, biometric systems for security applications are usually characterised by clients with good faith attempts to match their own template and by unintended impostor attempts (denoted as *zero-effort attempts*). Whenever impostors may easily imitate the required biometric data like e.g. behavioural data, impostor measures using *active impostor attempts* might be necessary [5]. Since defining the level of effort for active impostors is difficult, most performance evaluations deploy only zero-effort impostor attempts despite their inability to rate the threats in real applications.

The complete picture of the verification system requires to take two other errors into account: failure to enrol (FTE) and failure to acquire (FTA). The first error is caused by technological or user errors during the enrolment phase while the second error bases on sample acquisition errors when capturing the biometric sample.

Besides economical costs related with these errors, also ergonomical factors must be taken into account when judging or comparing different biometric systems. Especially the ease of use and the error handling procedures have a high impact on the perceived system performance and contribute thus to the public acceptance of a biometric technology.

2.4 Typical populations in behavioural based biometrics

Among different sources for variability of the matching results in biometric systems, the user plays the major role. Especially biometric technologies deploying behavioural data suffer from inherent differences of the recognizability in their population. For speaker recognition systems, Doddington extended in [8] the already used terms *sheep* and *goats* for characterising the recognition results of different *speaker types*.

- *Sheep* are the dominating part in the speaker population, their recognizability is predictable and recognition systems perform nominally well with them.
- *Goats* are denoted those speakers, who are difficult to recognize. They contribute to a larger part to the false rejections of a speaker recognition system. The reason for this is most often an erratic speaking style; especially children very often fall into this group.
- *Lambs* are denoted those speakers, who are easily imitated by others. Their contribution to the false acceptances is disproportionally high. Thus the knowledge about existing lamb speakers simplifies active intrusion into the system.
- *Wolves* are particular successful in imitating other speakers and contribute from the impostor side to a large part of false acceptances.

Only the recognition of goats and lambs are possible by a recognition system since only clients of the system have provided enough training data to build models. A comparison of error counts in [8] revealed that 25% of the most goat-like speakers contributed 75% of the false acceptances in a speaker recognition experiment. This uneven distribution of errors over the population can only be found with the goat speakers. In contrast, wolves and lambs show a nearly even contribution to the decision errors.

Several measures have been developed (see e.g. [9]) to detect problematic speakers during the enrolment phase and decide about additional processing steps like e.g. model retraining or score weighting.

Chapter 3

Speaker verification systems

In the last chapter we presented some fundamental properties of biometric systems using physiological and behavioural features. Now we proceed with a more detailed description of speaker verification systems and their state-of-the-art technologies. Most of the components of these systems were not uniquely developed for speaker verification but originate in general speech processing (e.g. the feature extraction) or in general decision taking systems e.g. the Bayesian approach for setting decision thresholds. We will concentrate on HMM (Hidden Markov Model) based speaker verification and refer to further modelling techniques found in the literature.

In the last section we will outline the MASV (Munich Automatic Speaker Verification) system that was developed as an experimental tool for our work.

3.1 Generalities

Verification versus identification

As we have noted already in the previous chapter, the verification task must be distinguished from the identification task of a biometric recognition system. In the context of speaker recognition both tasks turn out as follows:

- Speaker verification: test the hypothesis that a certain individual is the speaker of a given utterance. This leads to a two class test where the hypothesis can either be accepted or rejected.
- Speaker identification: find for a given utterance the speaker among a set of n registered speakers. This is typically a n class problem, if we are forced to select a speaker from the given population (denoted closed-set identification) or a $n+1$ class problem when we are allowed to assign the utterance also to an unknown speaker (open-set identification).

While an identification system depends on the number of registered speakers and typically shows decreasing performance with increasing speaker population, verification system do no suffer from this disadvantage. They are easy to scale up and are virtually limited only by the storage capacity of the database for the speaker models. However, a verification system needs an identity claim. Typically, a customer would type an ID number into the system to claim his identity. From an ergonomical viewpoint, this procedure does not offer an increased comfort for the user compared to traditional verification methods using e.g. PIN numbers.

A more natural way that fits seamlessly into speech dialog systems could be e.g. speaking one's name, in case that only a small speaker population is expected (see e.g. [10]). For larger speaker populations, an alternative could be using a proof of possession e.g. a smart card as identity claim. This will increase the security level for the verification process for both the system (combination of proof of possession and proof of property) and the user since the voice characteristics could be stored on the smart card instead in a central database.

Dependence on linguistic content

Verification systems can be characterised by the range of linguistic content of the clients' utterances. *Text-dependent* systems (also called fixed-phrase SV system) expect a pre-specified utterance that can be kept fixed for all clients or be chosen by the user. *Text-independent* systems are more flexible as any kind of linguistic content can be used during both enrolment and application phase. A kind of intermediate type are *text-prompted* systems where utterances are constrained by a fixed vocabulary. Commonly used are prompts of single digit strings or strings composed by two-digit numbers (spoken as e.g. "twenty-one").

As we will discuss later, text-independent systems are easier to implement but suffer also from the disadvantage of easier deliberate impostor attacks because any kind of spoken utterance can be used for verification. The same problem exists for text-dependent systems where a genuine utterance and a recorded one are very difficult to discriminate. Text-prompted systems try to tackle the recording threat by changing prompts with each access.

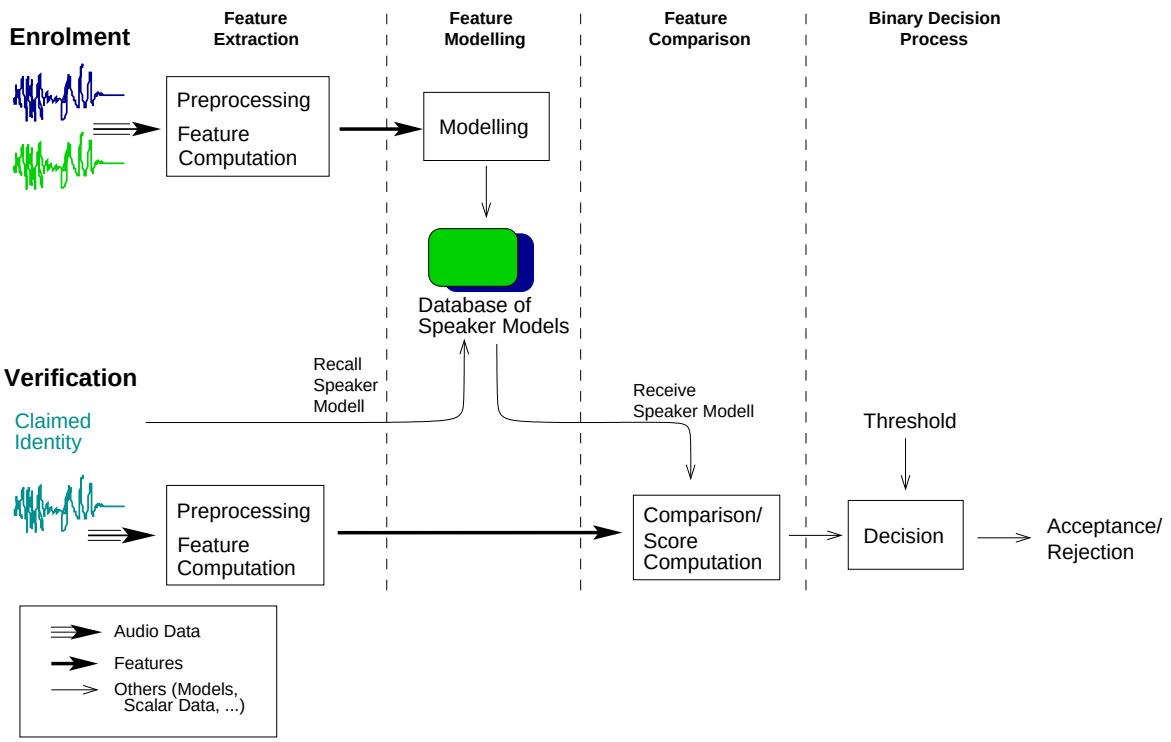

Figure 3.1: Schematic view of a speaker verification system. Processing steps are depicted for the enrolment phase (top) and the verification phase (bottom).

Basic setup of a speaker verification system

Figure 3.1 depicts the main components of a speaker verification system. Despite different techniques that can be used in each stage, the basic layout remains always the same. Similar to any biometric technique, the application of a speaker verification systems comprises two phases: the *enrolment phase*, where the systems captures the properties of a speaker and associates them with his identity, and the *verification phase*, when a speaker claims an identity and the claim is validated based on the claimant's speech sample.

The enrolment starts with the *feature extraction* stage where discriminating properties of speech are extracted and fed to the adjoining stage, the *feature modeling*. The model gives an enclosed description of a person's voice characteristics, based on the feature space generated in the extraction stage.

For the verification phase, the same feature extraction stage is applied to the speech sample of the claimant. In addition, the speaker claims an identity that should be proved. The speech sample and the claimant's model are compared in the *feature comparison* stage and a similarity measure, denoted as *score*, is calculated. In the final stage, the *decision process*, the similarity will be judged and results in the acceptance of the speaker or his rejection.

Each of the stages with typically deployed techniques are described in more detail in the following sections. We will mainly concentrate on speaker verification systems using *Hidden Markov Models* (HMMs) and refer to the literature for different approaches.

More detailed introductions and overviews can be found e.g. in [11, 12, 13, 14].

3.2 Feature extraction

Prior to any speech processing by a computer the speech signal must be converted from the analogue domain into a digital representation. Typically sampling rates of 16kHz and sampling resolutions of 16 bits are used in speech technologies. Whenever the speech originates from telephone networks, lower sampling rates are used, since modern speech coding schemes in telephony are designed for sampling rates of typically 8kHz. ISDN e.g. transmits speech with 8kHz and 8 bit A-law encoded samples.

Since the raw speech waveforms are not generalised enough to use them with a recognition system, the data rate is reduced by transforming the speech input into a sequence of feature vectors. The common assumption is, that speech can be regarded as stationary over segments of around 10-20ms. Thus, the feature vector sampling rate is much lower, around 100Hz.

Various features that have been developed over the last decades often originate from speech recognition research and were later also deployed for speaker recognition. Most of the successful features are calculated in the spectral domain and include filter bank analysis, linear prediction analysis and cepstral features. A good overview of features used in speaker verification can be found in [15].

At the end of section 2.1 we mentioned already the requirements for ideal features for biometric applications. Although none of the existing feature techniques can satisfy all these requirements, some of them are quite successful in speaker verification and identification. Ironically, they perform equally well in speaker-independent speech recognition and the speaker-dependent task of verification/identification.

Cepstral features are generally regarded as very successful representations for speech related recognition tasks. They provide a robust encoding of the spectral characteristics of speech both under clean and noisy conditions.

Since all of our experiments deploy cepstral features, we will describe them briefly the following. More details about the feature sets used through out in our experiments are listed in appendix C. For more information about feature extraction we refer to well known literature such as [4, 16, 17, 18].

Cepstral analysis

Originally, the *cepstrum* is computed using the logarithm of the power spectrum ([17]):

$$\hat{x}(q) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left(\log |X(e^{j\omega})| \right) e^{j\omega q} d\omega \tag{3.1}$$

The term $|X(e^{j\omega})|$ denotes the power spectrum of the signal $x(t)$.

In order to make the cepstrum amenable to computation, all involved Fourier transforms are replaced by their corresponding discrete variant, the DFT. Hence, the cepstrum is given as discrete sequence in the quefrency domain ($\hat{x}[q]$) with q denoting the cepstrum index.

An illustrative explanation of the meaning of the cepstral coefficients would be, that the zeroth coefficient $\hat{x}[0]$ describes the overall energy in the spectrum, while the first one $\hat{x}[1]$ gives the balance between the energy in the upper and the lower half of the spectrum. Higher order cepstra describe increasingly finer details of the spectral shape.

Two different approaches for computing cepstral coefficients have evolved which both differ from the original definition: mel frequency cepstral coefficients (MFCC) and linear predictive coding cepstra (LPCC). Despite differences in their definition, their main properties keep untouched.

Performance and robustness differences between MFCCs and LPCCs are presented later in chapter 6.

Mel frequency cepstral coefficients (MFCC)

The MFCC features $c^{\mathrm{MFCC}}[q]$ are given by

$$c^{\mathrm{MFCC}}[q] = \sqrt{\frac{2}{R}} \sum_{r=1}^{R} m_r \cos\left(\frac{\pi q}{R} (r - 0.5) \right) \tag{3.2}$$

where m_r denotes the logarithm of the energies computed from R spectrum bands on a Mel frequency scale.

Due to the property that the low-order cepstral coefficients describe the spectral shape, the first 12 to 13 coefficients are typically selected to create the feature vector. Commonly used window lengths are between 200 and 256 samples. Please note that several different variations of the DCT are deployed in formula 3.2, see [19, 17, 4]. We show here the formula used in HTK since we use HTK's preprocessing engine.

Linear predictive coding cepstra (LPCC)

An alternative to linear predictive coefficients (LPC, see e.g. [20]) for speech representation are cepstra based on these coefficients, denoted as linear predictive coding cepstra (LPCC). A simple recursive algorithm is given by Atal [21] in order to derive from p LPC coefficients $\{a_1, \ldots, a_p\}$ up to p LPCC coefficients $c^{\mathrm{LPCC}}[q]$:

$$c^{\mathrm{LPCC}}[q] = -a_q - \frac{1}{q}\sum_{k=1}^{q-1} k\, c^{\mathrm{LPCC}}[k]\, a_{q-k} \text{ with } q = 1, \ldots, p \tag{3.3}$$

Deployed database and feature sets

The VeriDat database, is described in more detail in chapter A. It gives an overview over the basic setup of this database which comprises speech from both fixed line and GSM telephone calls. We characterise shortly the speaker population, the speech items and the recording conditions. Additionally we refer to other speech databases for speaker identification/verification and compare them with the VeriDat database.

The basic feature sets used in our work are denoted as MFCC_SD and LPCC_SD. The HTK parameters for their computation are listed in appendix C. They comprise 12 cepstral coefficients and the total energy of a speech frame and are enriched by the velocity and acceleration coefficients. The extended versions of these basic sets include CMS normalisation and are denoted as MFCC_SD_CMS and LPCC_SD_CMS respectively.

3.3 Speaker models

Many different modelling techniques can be deployed for speaker recognition systems. In this section, we will briefly present the most common methods. Their applicability depends primarily on the type of the recognition/verification protocol: text-dependent systems allow the usage of different modelling techniques than text-independent systems.

Modelling techniques for text-dependent speaker recognition

In text-dependent recognition, the system expects a particular sentence spoken by the speaker. The linguistic content could be either fixed and directly associated with the speaker or could be generated by the system by prompting the user to make a specific utterance. In both cases, the speaker models can incorporate both the speaker identity and the linguistic content. Among the main methods are Dynamic Time Warping (DTW) and Hidden Markov Models (HMMs).

Dynamic Time Warping (DTW) This method is fairly simple to implement and has been used by many early speaker recognition systems. Each expected utterance of the speaker is stored completely by their representing sequence of feature vectors. During the recognition phase the currently spoken utterance is compared with the stored utterance by means of the dynamic time warping (see e.g. [22]) in order to compensate temporal variations of the speech segments. The computed distance between the two utterances is used later in the decision taking stage.

Hidden Markov Models (HMMs) HMMs are successfully deployed statistical models for many recognition tasks, where data sequences occur. A good introduction can be found in [22].

The model consists of a number of states which are each associated with a probability density function or a discrete probability function, denoted as emission probability functions. The states are connected with each other depending on the structure of the HMM. Typically for speech processing, left-to-right HMMs are deployed. These models consist of a sequence of states where each state exhibit transitions to itself and/or its successors.

During the training phase, the parameters such as the transition probabilities and the emission probability functions are estimated using learning algorithms such as e.g. the Expectation-Maximization (EM) [23]. In the pattern matching mode, the input sequence is mapped to all possible state sequences of the HMM, that could generate the input sequence. According to the emission probability functions, the total probability (for discrete emission probabilities) and accordingly the total probability density function (for probability density functions) can be calculated for the input sequence. For computational reasons, often not all possible state sequences are calculated in the matching mode but only the best path (Viterbi path).

For speech processing, HMM states are typically associated with distinctive segments of the speech signals. On a higher level, HMMs are mapped to phonetic or linguistic units such as phonemes, syllables, words or even whole sentences.

Due to the large amount of speech data required for training HMMs on sub-word levels, speaker recognition often use only word-level HMMs. Such a system could process any sequence of the trained words, e.g. different combinations of spoken two-digit numbers such as "twenty-one", "sixty-one" and so on. For fixed sentence recognition systems, a single HMM is build to capture the speaker's utterance.

The resulting probability and accordingly the resulting probability density function for a given utterance is processed later in the decision stage.

We refer for details of HMMs and the Expectation-Maximization algorithm to the literature: e.g. [24] provides a good introduction into the EM algorithm while the foundations of HMMs can be found e.g. in [25].

Modelling techniques for text-independent speaker recognition

In text-independent speaker recognition, the speaker does not need to utter the same sentence for each access. Thus, the model can capture the speaker based only on the acoustic characteristics and independent from the linguistic content of his utterance. The main methods for this type of speaker recognition include Vector Quantisation and Gaussian Mixture Models. Text-independent modelling techniques typically show a lower performance compared to text-dependent speaker modelling.

Vector Quantisation (VQ) Vector quantisation, described for instance in [26], uses a set of representative feature vectors, denoted as prototypes. The prototypes are computed by the LBG clustering algorithm (see e. g. [26]) in order to find distinctive areas in the feature space to separate speakers from each other. During verification of a given utterance, the sum of the distances between each feature vector and its closest prototype vector associated with the speaker is calculated. The total sum serves as a score to decide about the identity of the speaker.

Gaussian Mixture Models (GMMs) The Gaussian mixture models [27] are currently the most deployed modelling technique for text-independent speaker recognition. They are closely related to HMMs since they deploy a single state together with a emission probability density function defined by a mixture of a large number of Gaussians. Different phonemes of the speaker are now modelled by different regions of a single probability density functions. In contrast, the HMM approach uses several probability density functions in distinctive states for modelling the speaker's phoneme.

3.4 Score normalisation using a world model

Most currently deployed speaker modelling techniques like e.g. HMMs are based on statistical generative models. Their output is the conditional probability density function $p(\mathbf{O}|\lambda_i)$ that a given utterance $\mathbf{O}$ is generated by the model λ_i which is associated with speaker i. Using Bayes theorem, we can rewrite the later expression in order to get the *a posteriori* probability of the model λ_i for the given utterance $\mathbf{O}$:

$$P(\lambda_i|\mathbf{O}) = \frac{p(\mathbf{O}|\lambda_i)P(\lambda_i)}{p(\mathbf{O})} \tag{3.4}$$

For the final decision if the utterance has been produced by speaker i, the probability $P(\lambda_i|\mathbf{O})$ could be compared with a fixed threshold. However, the estimation of $p(\mathbf{O})$ is error-prone: undesired effects such as changing characteristics of the environmental and transmission channel noise or uncharacteristic speech sounds from the speaker have a great and also unknown impact on this pdf. Errors in this estimate result in increased false rejections and false acceptances.

A solution comes from the Bayesian decision theory that allows us to estimate the optimal decision threshold by minimising a cost function. In order to weight the occurrence of a false acceptance and a false rejection differently, the following cost function is defined [28]:

$$C = C_{FR}\ FRR\ P_{client} + C_{FA}\ FAR\ (1 - P_{client}). \tag{3.5}$$

Here FRR denotes the false rejection rate, C_{FR} the cost for a false rejection and P_{client} the a priori probability that a true client applies for verification. The parameters for the false acceptance case (FAR, $1 - P_{client}$ and C_{FA}) are defined similarly.

Its minimisation leads to the Bayesian decision rule (see e.g. [29]) that a claimed speaker i has to be accepted when

$$C_{FR}\, P(\lambda_i|\mathbf{O}) > C_{FA}\, P(\bar{\lambda}_i|\mathbf{O}). \tag{3.6}$$

where $P(\bar{\lambda}_i|\mathbf{O})$ denotes the probability, that any other speaker than i has produced the given utterance.

Together with formula 3.4, we can rewrite this condition as likelihood ratio test:

$$\log p(\mathbf{O}|\lambda_i) \ - \ \log p(\mathbf{O}|\bar{\lambda}_i) \ > \log \frac{C_{FA}\, P(\bar{\lambda}_i)}{C_{FR}\, P(\lambda_i)} \ = \ \theta_i. \tag{3.7}$$

Here we applied the logarithm on both sides of the relation, which leads to a *log likelihood ratio* computed from both scores. The Bayesian decision threshold θ_i is theoretically defined by formula 3.7. If the true costs can not be determined, the equal cost approach ($C_{FR} = C_{FA}$) is often deployed. Further aspects of the Bayesian decision threshold are discussed in the following section.

The final result of the normalisation stage is the normalised score S_i as suggested by formula 3.7:

$$S_i = \log p(\mathbf{O}|\lambda_i) \; - \; \log p(\mathbf{O}|\bar{\lambda}_i). \tag{3.8}$$

Several different approaches exist to estimate or generate an *anti-speaker model* $\bar{\lambda}_i$ describing all other speakers. In the simplest case, we assume that this model is the same for all client speakers and we can replace it by a general client-independent model Ω. It is often denoted as *universal background model (UBM)*, simply *background model* or *world model*. Typically this model has the same topology as the client speaker models and is trained by pooling data from an extra set of speakers, denoted as the *world speakers*.

Other approaches [30, 31, 32, 33] use client-specific anti-speaker models that are based on a cohort of speakers whose models are most competitive with the client model. Among these techniques is the selection of the cohort speakers from the client speaker pool [30] or the construction of a virtual cohort model for the current client model based on the closest Gaussians mixtures from the remaining client models [32]. All cohort-based techniques report performance improvements compared to the standard world model approach.

3.5 Decision taking

In the previous section we already presented the Bayesian decision threshold for the normalised score

$$\theta_i = \log \frac{C_{FA}\, P(\bar{\lambda}_i)}{C_{FR}\, P(\lambda_i)}. \tag{3.9}$$

By using the normalised score S_i for a given identity i (see formula 3.8), we can easily decide about the claimed identity:

$$\begin{aligned} S_i > \theta_i \quad &: \quad \text{accept speaker } i \\ S_i \leq \theta_i \quad &: \quad \text{reject speaker } i \end{aligned}$$

However, this threshold is theoretically defined; the true probabilities $P(\lambda_i)$ and $P(\bar{\lambda}_i)$ are unknown and must be estimated *a priori* from additional speech data. In addition, the pdfs $p(\mathbf{O}|\lambda_i)$ and $p(\mathbf{O}|\bar{\lambda}_i)$ can not be achieved directly but are often estimated based on the likelihood function of statistical models such as HMMs.

Thus, changing conditions like e.g. the access probability of a client speaker, the handset type, the speaker's mood and health state introduce additional score variability. Further score normalisation techniques have been developed to address these difficulties.

Instead of applying a speaker-independent decision threshold on the world normalised scores (see formula 3.9), speaker-dependent thresholds or fixed thresholds with rescaled scores are often deployed.

Among the most well known normalisation techniques is the Z-normalisation [34] (Z-norm) that is based on earlier work of Furui [18]. Here the impostor score distribution for each client speaker is assumed to be Gaussian. The speaker-dependent score bias μ_i^{imp} and score spread σ_i^{imp} are estimated using a set of development impostors. Then the speaker's scores are rescaled in order to receive a standard normal distribution:

$$\hat{S}_i = \frac{S_i - \mu_i^{\text{imp}}}{\sigma_i^{\text{imp}}} \tag{3.10}$$

Setting a fixed threshold on the rescaled score $\hat{S}_i$ actually sets an equal false acceptance rate for all client speakers.

Further score rescaling methods are based on the Z-norm approach. H-norm [35] calculates normalisation parameters for each speaker and each handset used during the tests. While H-norm calculation is performed during the training phase, T-norm (Auckenthaler et al. [36]) feeds a verification utterance to the non-target speaker models in order to calculate the normalisation parameters.

3.6 The MASV system

MASV stands for "Munich Automatic Speaker Verification". It is an experimental environment for performing speaker verification (SV) that allows an easy set up of SV systems using different kinds of models including sub-word-model HMMs and GMMs. MASV is published under version 2 of the GNU General Public License. It is based on the tools of HTK (Hidden Markov Model Toolkit) [4].

Overview

MASV comprises two parts: a Perl script part (based on the HTK tools) which handles training and testing of an experimental SV system and a Matlab script part, which performs evaluation and performance measurement of the system. Information from the Perl part to the Matlab part is transferred by text files (MLF files of HTK and so-called *info files* describing the training and evaluation configuration including e.g. lists of speakers used as clients, number of iterations for a model, etc.).

There are two types of Perl tools, denoted as high level and low level tools. The high level tools allow an easy configuration of the SV system under test and the generation of a master bash shell script. This shell script calls the low level Perl scripts which act as wrapper scripts for the several HTK tools including HCompV, HRest and HERest (model training), HHEd (model manipulation) and HVite (evaluation). Beside preparing options and script files for the HTK tools, these Perl scripts provide the mechanism for the distributed execution of the training and the evaluation on several hosts.

The second part of MASV is a bunch of Matlab functions which perform further processing of the data produced in the first step. Here, HTK's MLF files and the info files are read and the performance of the system is computed. The final results are stored in a special Matlab data type, a structure, which is stored in a file. In addition, several graphical user interfaces allow exploring the experiment data and generating various plots.

The whole system is developed for the German VeriDat Speaker Verification database which will be presented in more detail in chapter A. Different databases can be used with MASV, however, the general file structure of the VeriDat database must be provided. The adaptation of MASV to other speech databases can be easily done since all database specific functions and description files are grouped into a dedicated script directory.

The latest version of the MASV-package including documentation is available from `http://www.phonetik.uni-muenchen.de/Bas/SV/` [37].

Requirements

MASV uses HTK (`http://htk.eng.cam.ac.uk`) for training and testing the modules. It is developed to work with version 3.1 of the HTK tools but newer versions should also be fine. There are some patches available (see the instructions on the MASV home page [37]) which arose from the development of MASV.

MASV requires Perl of version 5 or higher, the bash shell and Matlab (version 5 or higher). Some parts of the code run currently only on UNIX compatible operation systems because of the usage of command line tools like e.g. `ssh` and `ln`. Two additional Matlab packages are required by MASV, `voicebox` by Mike Brookes and `matdraw` by Keith Rogers.

Key features

Apart all speaker verification techniques used in this work, MASV offers several additional features. We summarise the key features in the following:

- distributed computation for both training and evaluation of an experimental setup.
- composition of speaker sets and training/evaluation items easily changeable.
- easy selection among pre-computed acoustic feature sets.
- speaker modelling by sub-word HMMs, whole-phrase HMMs, GMMs and speech rate templates.
- speaker models generated either from flat start or by MAP adaptation of existing models (world models, gender-dependent models).
- score normalisation by world normalisation, simple cohort normalisation (mean score of n-best cohort speakers or maximum score of cohort speakers) and handset normalisation (H-norm).
- selection among several background models based on additional information about an utterance (e.g. level of background noise, telephone-type).
- evaluation by dedicated client/impostor sets or by leave-one-out protocol.
- different threshold setting approaches including a-priori threshold for equal error rate calculated on dedicated development speakers set.
- calculation of various performance measures including DET plots and their confidence bounds (see chapter 4).
- Matlab GUI for exploring and comparing results, filtering evaluation data (based on speaker identity, speaker sex, recording type) and creating performance plots.

Chapter 4

Performance evaluation

In section 2.3 we already mentioned some general aspects of the performance of a biometric system. This chapter will present the established performance criteria used in speaker verification research. We will start with typical sources of variability for performance measures that occur in speaker verification systems. After that, we present common performance parameters that have been established over the last years. Following, we will deal with various approaches of estimating confidence bounds for the performance parameters. Finally, we present construction techniques of the confidence intervals for both the *Detection Error Trade-off* curves (DET curves) and the *Equal Error Rate* (EER).

4.1 Sources of variability of performance measures

Measuring performance of any technical system requires appropriate data for predicting its service quality during application in real life.

Generally, the quality of the training and evaluation data and the conditions under which the data are collected influences the performance measure estimation. Poor quality of the data may produce results not reflecting the true performance of the system in its later application. Similarly, too optimistic quality of the data will give overestimated performance figures. Test and training data must therefore match as closely as possible the normal operating conditions. In addition the quality between training and test data shall be comparable in order to avoid a bias of the performance figures.

The mentioned quality of the data is affected by various sources of variability:

Speaker population

The main source of variability is introduced by the speakers themselves. Their age, gender and physiologic properties e.g. physiology of the vocal tract play an important role. Disabilities or illness can introduce permanent or temporal variations of their voice.

The client's behaviour is another important aspect for voice verification as speech is influenced by long-term variations e.g. dialects, accents or intonation and also short-term variability introduced by the emotional state of the speaker.

Environmental influences

In case of voice verification background noises or other voices are the main influence from the environment. It alters the recorded signal and can also affect the ability of the user to hear the instructions.

Hardware influences

Differences in the microphones and handsets used can be mainly accounted to this point. The transmission channel can add noise and corruptions to the signal. It can vary between trials and even show short-time variations due to load balancing in the networks.

4.2 Established performance parameters

Test performance is an important aspect of any biometric identification or verification system. In most cases, it is considered impossible to perform a test with all subjects of the target population. Thus it is necessary to make statements about the entire intended population using a population sample.

Two very good introductions on common parameters for describing the performance of a biometric system can be found in [5] and [38]. As we focus here on a speaker verification system we will concentrate on the relevant parameters for this type of biometric system. More elaborate variations of the parameters presented here are described in [29].

4.2.1 False rejection and false acceptance

For speaker verification, two types of error must be distinguished. *False rejection* (FR) occurs when a genuine speaker is rejected, *false acceptance* (FA) occurs when an impostor is accepted as the speaker he claimed he was (the *violated speaker*). Both errors are counted over a set of utterances and their proportion to the total numbers of tests give an estimate of the *False Reject Rate* (FRR) and the *False Accept Rate* (FAR). Both error rates can be written as a conditional probability:

$$\begin{aligned} FRR &= P(\text{ reject } | \text{ client present }) \\ FAR &= P(\text{ accept } | \text{ impostor present }) \end{aligned} \tag{4.1}$$

Very often a biometric system is characterised as a hypothesis test and the occurring terms are denoted in terms of hypothesis testing. We define H_0 and H_1 as:

$$\begin{aligned} H_0 &: P = C, \text{ the person is the client} \\ H_1 &: P \neq C, \text{ the person is an impostor} \end{aligned} \tag{4.2}$$

Thus both errors can also be denoted as "*Type I*" error (rejection of a true hypothesis H_0, equivalent to FRR) and "*Type II*" error (accepting a false hypothesis H_1, equivalent to FAR). Other equivalent terms for the FRR are *false positive rate* or *false alarm rate*. Similarly for the FAR the terms *false negative rate* or *miss rate* are found in literature.

Both error types can only be estimated using a test set that should match as closely as possible the conditions used in operating mode. For the purpose of evaluation both the training and test material should be therefore collected under the same conditions. We will distinguish between the true value of a measure x and its estimated value $\hat{x}$. The notation scheme follows closely [29].

Scoring notation

A registered speaker X_i produces g_i genuine test utterances. The index i runs from 1 to m, the total number of registered speakers known to the system. In order to simplify the notation, we assume that each registered speaker is tested with the same amount of genuine test utterances g, i.e. $g_i = g \,\forall\, i$. The set of one speaker's test utterances is denoted $\boldsymbol{x_i}$:

$$\boldsymbol{x_i} = \{x_i^1 \dots x_i^g\} = \{x_i^k\}_{1 \le k \le g} \tag{4.3}$$

The total number of tests with genuine utterances is denoted as G:

$$G = \sum_{i=1}^{m} g_i = mg. \tag{4.4}$$

A set of impostor utterances can be divided in subsets corresponding to one of n impostors $\{Y_j\}_{1 \le j \le n}$ using the system with the claimed identity X_i. Again, we assume, that each speaker provides the same number of utterances i.e. each impostor performs the same amount of impostor test utterances $h_j = h$. The set of utterances of impostor j claiming identity i is

$$\boldsymbol{y_{ji}} = \{y_{ji}^1 \dots y_{ji}^h\} = \{y_{ji}^k\}_{1 \le k \le h}. \tag{4.5}$$

Practically the same utterances of an impostor can be used for any claimed identity. The notation defined above using explicitly the index of claimed identity facilitates the definition of the false acceptance rate later on.

Similarly to the total number of tests with genuine utterances (4.4), we will denote as H the total number of tests with impostor utterances:

$$H = \sum_{i=1}^{m} \sum_{j=1}^{n} h_j = nmh. \tag{4.6}$$

A verification system can be viewed as a function $v(i, z)$ which assigns a Boolean value to a given test utterance z with the claimed identity i:

$$v(i, z) = \begin{cases} 1 & : \text{ identity } i \text{ accepted} \\ 0 & : \text{ identity } i \text{ rejected} \end{cases} \tag{4.7}$$

Using the function $v(i, z)$ simplifies a common notation of the FRR and the FAR.

False Reject Rate

A false rejection corresponds to:

$$v(i, x_i^k) = 0 \tag{4.8}$$

The false reject rate is defined by the amount of false rejection errors in proportion to the total number of genuine test utterances. This rate can be calculated for an individual speaker, a subset of speakers (e.g. male/female clients) or the complete client population. In the latter case, we define the estimate of the overall $\widehat{FRR}$ as

$$\widehat{FRR} = 1 - \frac{1}{G} \sum_{i=1}^{m} \sum_{k=1}^{g} v(i, x_i^k) \tag{4.9}$$

The same error rate can be defined for a client i individually:

$$\alpha_i = 1 - \frac{1}{g}\sum_{k=1}^{g} v(i, x_i^k) \tag{4.10}$$

In terms of the terminology of Doddington's zoo, clients with high α_i i.e. frequent rejection of their genuine trials are called "goats" while clients with low α_i are denoted as "sheep".

False Accept Rate

Similarly to the FR, a false acceptation occurs when:

$$v(i, y_{ji}^k) = 1 \tag{4.11}$$

For the definition of the false accept rate there are several ways to score the false accept rate ([29], Chap. 11.4.3.2). For the overall false accept rate, we score non-genuine trials globally, regardless of the impostor identity nor of the claimed identity:

$$\widehat{FAR} = \frac{1}{H}\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{h} v(i, y_{ji}^k) \tag{4.12}$$

A more detailed analysis of the FAR is presented in [29]. Beside the overall FAR, an average FAR in favour for an impostor Y_j (denoted as $\tilde{\beta}_j$) and an FAR against speaker X_i (denoted as β_i) is defined.

The figure β_i indicates the false accept rate obtained on average by each impostor Y_j in claiming identity X_i. A registered speaker with a low β_i can be viewed as being resistant, or, being vulnerable having a high β_i. In Doddington's zoo terminology these two sorts of clients are called *lambs* (high β_i) and *rams* (low β_i) respectively.

Looking at $\tilde{\beta}_j$, we find poor impostors with low $\tilde{\beta}_j$ (called *badgers*) and skilled impostors with high $\tilde{\beta}_j$ (called *wolves*).

We will come back to this different behaviour among clients in section 2.4.

General comments on FRR and FAR

There exist several different estimates of the global error rate, using different assumptions on the relative representativity of each speaker. In the definitions used here it is assumed that each speaker has a representativity which is proportional to its number of test utterances. Using the test protocols defined on page 40 and 41 we selected test and training data in regard with an equal number of utterances per speakers. In addition the speaker sets were defined nearly gender-balanced (see page 38). If the composition of the speaker population is not representative with regard to gender, gender-balanced scores calculated from the mean of error rates of the female speaker subset and the male speaker subset provided a reasonable estimate. We have found in our experiments that the error rates based on 4.9 and 4.12 are very close to the gender balanced values. More sophisticated balancing schemes are given in [29].

Same-gender impostors

Many biometric systems uses partitioning where only samples and templates are compared that fall in the same bin. The bins are defined e.g. in a fingerprint system by the same "arch/loop/whorl" type or in a speaker verification system by the speaker gender. Within-bin comparisons show typically higher FA rates than comparisons using all available impostor data. In order to show this effect, we will report also same-gender FA rates. The definition of the FAR (4.12) is slightly extended to

$$\widehat{FAR}_{MF} = \frac{1}{H}\left(\sum_{\substack{i=1\\ i\in\mathcal{M}}}^{m}\sum_{\substack{j=1\\ j\in\mathcal{M}}}^{n}\sum_{k=1}^{h} v(y_{ji}^k) + \sum_{\substack{i=1\\ i\in\mathcal{F}}}^{m}\sum_{\substack{j=1\\ j\in\mathcal{F}}}^{n}\sum_{k=1}^{h} v(y_{ji}^k)\right) \tag{4.13}$$

$\mathcal{M}$ and $\mathcal{F}$ are the sets of male and female speakers respectively. Generally it is assumed that *casual* impostors are more successful in violating a same-gender speaker than a cross-gender client.

4.2.2 Threshold setting

In general, biometric verification systems proceed in two steps. First, a score $s(z, X_i)$ between a sample z and a reference model X_i is computed. Second, the value of the matching score is compared to pre-defined threshold θ_i and a decision is taken as follows:

$$\begin{array}{lcl} s(z, X_i) \geq \theta_i & \Rightarrow & v_i(z) = 1 \\ s(z, X_i) < \theta_i & \Rightarrow & v_i(z) = 0 \end{array} \tag{4.14}$$

In other words, the verification is successful if the match between the utterance z and the reference model X_i is close enough.

A very similar decision rule can be used if the matcher outputs a distance between sample and model rather than a score. In this case a weak match will result in a high distance and hence the inequality signs have just to be reversed.

The notation for the threshold θ_i implies that a *speaker-dependent threshold* is allowed in the most general case. This approach allows a specific adaptation to the characteristics of a single speaker. However the problem remains in choosing the right threshold per client. More commonly used is a *speaker-independent threshold* where each threshold is set to the value θ.

Choosing the right value for θ_i is a kind of tradeoff as it has a inverse impact on the FRR and FAR. A low θ_i will lead to fewer genuine attempts being rejected, but more impostors will be erroneously accepted and vice-versa. Figure 4.1 illustrates the opposed behaviours of FRR and FAR.

Among many possibilities, the conditions for setting the threshold θ_i can be:

- Achieving a specified false reject rate FRR_0. Often applied in low-security systems where customer satisfaction is more important than fraud protection.
- Achieving a specified false accept rate FAR_0. This case applies to the inverse situation of the previous point. A high level of security against imposture is in the main focus.
- Achieve the same value for both errors. The *equal error rate* (or *EER*) is explained more detailed in the next subsection.

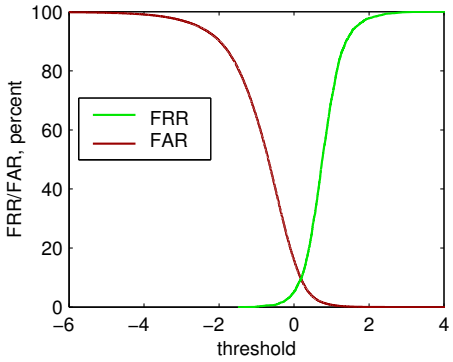

Figure 4.1: Characteristics of FRR and FAR vs. decision threshold. Example taken from a speaker verification system using GMMs (four mixtures, feature set MFCC_SD, training and test protocol "separate impostors").

- Achieve the least expected costs. If P_{client} denotes the probability of a genuine client attempt, C_{FR} gives the costs for a false rejection and C_{FA} gives the cost of a false acceptance, the expected overall costs are given by the *decision cost function* (*DCF*)

$$C = C_{FR}\ FRR\ P_{client} + C_{FA}\ FAR\ (1 - P_{client}). \tag{4.15}$$

  In particular, when $P_{client} = 0.5$ and both costs are equal ($C_{FR} = C_{FA} = K$), the *equal-risk equal-cost expected costs* are

$$C_{eq} = K(1 - (FRR + FAR)). \tag{4.16}$$

4.2.3 Equal error rate (EER)

In the past researchers used the EER criterion for reporting performance of their laboratory systems. It is a simple way of summarising the overall performance in a single figure. In most practical applications however, the equal error rate is not suited for description of the in-field performance, because these applications normally do not operate at the EER point. More often, threshold settings with pre-defined FRR or FAR are used. Additionally the equal error rate can only be estimated with an a posteriori threshold setting. In this case the EER of the system must be understood as the performance with ideal thresholds.

Using a priori thresholds requires tuning data which can be the training data itself or a separate data set. A sound experimental procedure requires that none of the impostors and none of the speakers in the tuning data set intersects with the corresponding speaker group in the test set. After setting the appropriate threshold the test set can be used to estimate FRR and FAR. Naturally, the final performance figures will differ from the performance achieved on the tuning set.

4.2.4 ROC and DET curves

In order to compare different biometric systems we need a more informative criterion than just a single performance figure. We can make use of a curve generated by plotting both types of errors at all possible threshold settings. The system's behaviour can thus be characterised independent from a fixed operating point. There are two common used plots of this type for describing the performance of imperfect diagnosis, signal detection or pattern matching systems:

the *Receiver Operating Characteristic* (*ROC*) and a modified ROC, the *Detection Error Trade-off* (*DET*) curves [39]. The ROC curve plots, parametrically as a function of the decision threshold, the "false positive" rate (here: FAR) on the abscissa against the corresponding "true positive rate" (here: 1 - FRR) on the ordinate.

The DET curve can be understood as slight variation of the ROC curve. It plots the "false negative" rate (here FRR) on the ordinate giving a reversed trend compared to the ROC plot. DET plots are therefore more intuitive than ROC plots since both rates are directly displayed.

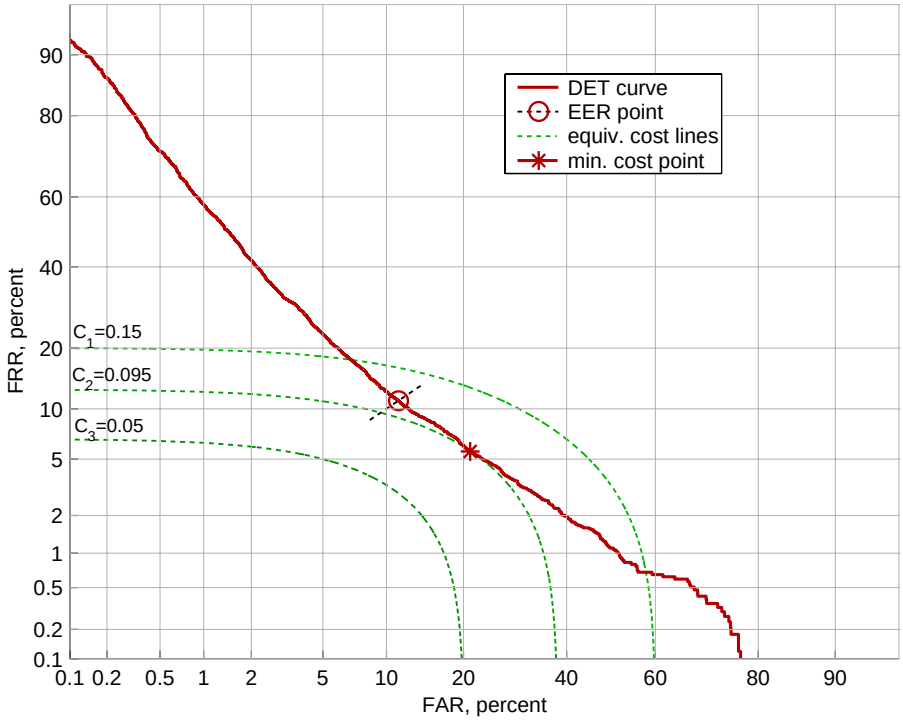

Figure 4.2: Example of a DET curve. EER point and operating point with minimum DCF. Equal cost lines for an arbitrary example using $C_{FR} = C_{FA} = 1$ and $P_{client} = 0.75$. Example taken from experiments with a GMM SV system (four mixtures, feature set MFCC_SD, training and test protocol "separate impostors").

Most often the curves of different verification systems are very close to each other. Plotting the graph using logarithmic axes or a *normal quantile-quantile plot* (*nqq-plot*) spreads the plot and distinguishes different curves more clearly. All depicted DET graphs used here are plotted using the nqq-plot. As elaborated in [39], normally distributed genuine test scores and impostor test scores result in straight lines in the nqq-plot. The slope of the lines is determined by $-\sigma(I)/\sigma(C)$, where $\sigma(I)$ denotes the standard deviation of the impostor scores and $\sigma(C)$ denotes the client scores. Equal standard deviations result in a negative unity slope of the DET curve. This property can also be interpreted locally when the normality assumption of the scores is not true. E.g. a local slope with an absolute value below unity indicates that the client scores locally spread out more than the impostor scores.

The point-to-point knowledge of the DET curves provides easy determination of the required threshold for different operating conditions listed in 4.2.2. Especially the equal error rate point is found at the intersection of the DET curve with the straight line defined by $FAR = FRR$. Figure 4.2 depicts an example DET plot with the EER point and an operating point according to a predefined cost constraint.

The EER can be interpreted as a very local property of the DET curve. In contrast the DET itself offers a complete description of the system under the selected operating condition but suffers from a desired level of conciseness. Oglesby suggested a simple 1-parameter model which allows capturing the system performance into a single number. Please refer to [29] for more details.

4.3 Analytical confidence bounds

The parameters presented in the previous section are inherently statistical statements. Usually these assessments form the basis for predicting the system performance on the entire population. As with any statistical parameter the question arises how accurate an estimate is. The two issues in this context are test size determination and confidence interval creation. Selecting the right amount of test data is a crucial point when a biometric system has to be evaluated with regard to specified requirements since the test data size determines the level of confidence for the estimated parameters. The final goal is a statistical sound statement about a comparison of two different systems or about the compliance of a system with its requirements.

As acknowledged in several papers (e.g. [40, 5]) there is a pressing need for estimating the confidence bounds of the performance parameters. The most important figures for estimating performance are the FRR, the FAR, and the closely related DET curves. Over the last years several different methods were developed for estimating confidence bounds of these parameters.

Though, at present there is no widely accepted method for creating a confidence interval. In this section we will present the different approaches and discuss their advantages and disadvantages.

McNemar test

The McNemar test is a χ^2 test for goodness of fit for two dependent samples and alternative data. The two samples are the decisions of two different biometric systems A and B which are each divided into two parts, the correct decision and the false decisions. Both systems are tested with the same test set, thus we have two *dependent* samples. A^+ and A^- indicate the events of correct and incorrect decisions for system A. We denote with $n(.)$ the frequency of these events and combinations between them. The McNemar matrix is defined as

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} = \begin{bmatrix} n(A^+ \cap B^+) & n(A^+ \cap B^-) \\ n(A^- \cap B^+) & n(A^- \cap B^-) \end{bmatrix} \tag{4.17}$$

The null hypothesis is that both systems perform equally; the corresponding test statistic for that case is

$$\chi^2 = \frac{(M_{12} - M_{21})^2 - 1}{M_{12} + M_{21}} \tag{4.18}$$

Note that this test implies the condition $M_{12} + M_{21} \geq 20$. More precisely, it is a modified version of the McNemar test for use with small values of the denominator in 4.18, known as *corrected McNemar test*. A difference between both systems is significant when

$$p = 1 - F(\chi^2) < \alpha \tag{4.19}$$

with F as the cumulative distribution function of the χ^2-distribution with 1 df and α as the confidence level.

The McNemar test is computational very easy to perform but does not distinguish between the FRR and FAR. It can give only a global impression of the differences between two biometric systems.

Binomial bounds

A common model for estimating FRR and FAR is a Bernoulli experiment with independent trials from a population of decisions. Further, it is assumed that the decisions are independent from each other and that the error probability is constant over the population of decisions. The assumed properties of the decisions must also be reflected by the scores i.e. the scores must be stochastically independent and have a constant distribution. This property is denoted as *independent identically distributed* (*i.i.d.*). We will discuss the shortcomings of these assumptions later on page 31.

Using this model, estimation of confidence intervals and required size of test data becomes fairly easy. In the following the binomial model is presented for estimating the FRR. Estimating the binomial bounds of the FAR is done similar.

Our objective is to estimate the error probability p by counting "successes", i.e. false rejections made by the system, and by calculating the frequency of successes based on the total number of tests. Let Y denote a random variable that represents the successes in n total trials. The frequency of successes, an estimate for p, is therefore also a random variable denoted by $\hat{p}$. Following our assumption, Y is a binomial random variable $b(n, p)$ with real but unknown value p as the error rate. The probability that Y is within a certain range $[y_1, y_2]$ is

$$\Pr\{y_1 \leq Y \leq y_2\} = \sum_{y=y_1}^{y2} \binom{n}{y} p^y (1-p)^{n-y}. \tag{4.20}$$

When n is sufficiently large ($n \geq 30$) and p neither to small nor too large, the binomial distribution can well be approximated by a normal distribution. A more explicit criterion for the validity of this approximation is [41]

$$n > \frac{9}{p(1-p)}. \tag{4.21}$$

Even small errors can be handled if n is sufficiently large (e.g. $p = 0.001 \quad \Rightarrow \quad n > 9000$).

Using the normal distribution approximation we can write 4.20 as

$$\begin{aligned} &\Pr\{y_1 \leq Y \leq y_2\} \approx \\ &\quad \approx \Pr\left\{ \frac{Y}{n} - z_{\alpha/2}\sqrt{p(1-p)/n} \;\leq\; p \;\leq\; \frac{Y}{n} + z_{\alpha/2}\sqrt{p(1-p)/n} \right\} = \\ &\quad = 1 - \alpha \end{aligned} \tag{4.22}$$

The term $z_{\alpha/2}$ indicates the number of standard deviations from the origin required to encompass a proportion $\alpha/2$ of the area under the standard normal distribution. For $\alpha = 5\%$, this value is 1.96.

As the true error probability is unknown we replace p with its estimate Y/n. The resulting confidence interval is

$$\left[(Y/n) - z_{\alpha/2}\sqrt{(Y/n)(1-(Y/n))/n} \quad , \quad (Y/n) + z_{\alpha/2}\sqrt{(Y/n)(1-(Y/n))/n} \right]. \tag{4.23}$$

The required sample test size is given as

$$n \approx \left(\frac{z_{\alpha/2}}{\epsilon}\right)^2 (Y/n)(1-(Y/n)), \tag{4.24}$$

where ϵ denotes the half width of the confidence interval.

Simplified binomial bounds: "Rule of three" and "Rule of 30"

Several commonly used approximations are derived from the binomial approach [5]. The "Rule of three" estimates a confidence interval when no errors are observed. The "Rule of 30" is a simplified version of the test size estimation while it puts its main focus on the number of errors (in our notation: successes) that have to be observed for a 90% confidence interval.

Rule of three

The "Rule of three" addresses the question "What is the lowest error rate that can be statistically established with a fixed number of n trials?" [42]. This question translates to the problem "What is the worst possible scenario for p when no errors are observed?". Again we assume that the trials are i.i.d. attempts.

From 4.20 we find

$$\Pr\{Y = 0\} = (1-p)^n. \tag{4.25}$$

The $(1-\alpha)$ 100% upper bound for p is found by calculating the least upper bound for p_u:

$$\begin{aligned} p_u &= 1 - \alpha^{1/n} \\ &\approx -\frac{\ln(\alpha)}{n}. \end{aligned} \tag{4.26}$$

The interval $(0, p_u]$ provides then $(1-\alpha)$ 100% coverage for the true p. For a 95% confidence level ($\alpha = 0.05$, $-\ln(\alpha) \approx 3$) we find the final rule

$$p_u \approx \frac{3}{n}. \tag{4.27}$$

Using e.g. one of our test protocols with 3360 independent genuine speaker tests with the assumption that no false rejections occurred, it can be said with 95% confidence that the true FRR is 0.9‰ or less.

Rule of 30

Doddington [43] proposed the "Rule of 30" that facilitates determination of the required test size. It is also known as "Doddington's law":

> To be 90% confident that the true error rate is within ± 30% of the observed error rate, there must be at least 30 errors.

The derivation of this rule starts again with the binomial assumption and the derived confidence interval 4.23. If p is small, $1-p$ is near 1 and interval simplifies to

$$\begin{aligned} &\left[(Y/n) - z_{\alpha/2}\frac{\sqrt{Y}}{n} \;,\; (Y/n) + z_{\alpha/2}\frac{\sqrt{Y}}{n}\right] \\ &\quad = [(Y/n)(1-f) \;,\; (Y/n)(1+f)] \quad \text{with} \quad f = z_{\alpha/2}/\sqrt{Y}. \end{aligned} \tag{4.28}$$

The factor f translates to the proportion of the size of the confidence interval with regard to the estimated error rate $\hat{p}$. Using $\alpha = 0.10, z_{\alpha/2} = 1.65$ gives together with $f = 30\%$ the minimum number of required errors is approximately 30. As we usually define the confidence level here with 95%, the number of required errors increases to about 43. In

our plots we will use the reverse way and calculate confidence intervals by estimating f at the observed number of errors. However, we will still refer to the "Rule of 30" as an approximation of the binomial distribution.

Confidence area based on the binomial assumption

Combination of the lowest absolute error rates that can be estimated (rule of three) and the confidence bands estimated with the binomial approximation (exact or approximated with the rule of 30) provides a confidence area for the DET curve. Figure 4.3 depicts this region. Not displayed are the upper absolute bounds for error rates close to unity, that can be constructed using the same argument deployed for the rule of three. However, most of the time, the area with low error rates is of interest.

In our following plots, we will always show only the confidence bounds and discard the lower absolute bounds of the error rates. Since we apply different techniques for estimating the confidence bounds, the combination of the confidence areas with the absolute bounds based on the rule of three would not be consistent. However, the general idea of figure 4.3 that an absolute lower bound for the error rates exists remains valid.

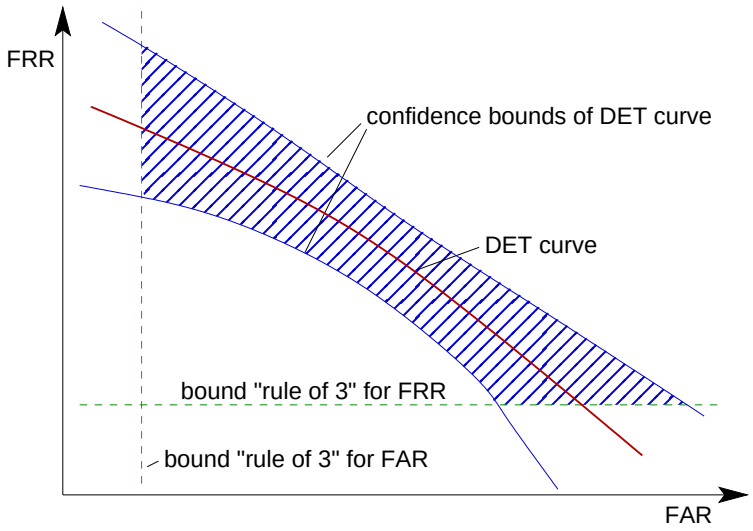

Figure 4.3: Schematic view of the confidence area based on the binomial assumption. The band is bounded to low FRR and low FAR rates by the rule of three.

Confidence bounds based on sub-sampling type 1

Several authors (see e.g. [40]) state that the binomial distribution gives inaccurate estimates of the real variance in biometric systems. While the variance of the FR rate is underestimated, the variance of the FA rate is grossly overestimated by the binomial distribution. This happens due to the binomial requirement of stochastically independent impostor test trials. Thus, N (the total number of trials) renders equal to the number of speakers in a cross-comparison impostor test.

In order to achieve small confidence intervals while holding the binomial assumption, we would require huge amounts of participating persons in the data collection.

Common practice is acquiring multiple trials per person. It is significantly easier to collect multiple utterances from the a small number of clients than collecting few utterances from a large number of participants. Moreover, whenever a recording session is made, it is easier to record more trials in a single sessions than splitting the data collection to multiple sessions.

User behaviour may vary with each successive attempt due to the possible factors already mentioned in section 4.1. As a general rule ([5, section 3.4]) the number of people is more significant for determining accuracy of error rates than the number of attempts. Collecting data in multiple sessions on different days helps to reduce the dependencies between samples of the same person. In turn, this reduces the errors introduced by a too optimistic assumption of stochastic independence as it is the case for the binomial assumption.

The dependency of the trials within a single speaker can be taken into account by assuming an underlying statistical model of the scores. An estimation of the variance for both FRR and FAR is given in [5] and [40] and will be presented here briefly. Estimating the confidence bounds of the FAR requires in this context the application of a cross-validation scheme.

FRR

For the FRR the situation can be tackled with a known method to calculate the variance of an estimated parameter when drawing samples from a base population. Each sample can be regarded as a person in our case and provides a subsample, here a collection of trials. Let n be again the number of clients and g_i the number of test trials of each client. The estimated variance for $\widehat{FRR}$ is given e.g. by [44, section 21.5]:

$$\begin{aligned} \hat{V}(\widehat{FRR}) &= \frac{\sum_{i=1}^{n} g_i(\widehat{FRR}_i - \widehat{FRR})^2}{\bar{g}^2 n(n-1)} \\ &= \frac{\sum_{i=1}^{n} a_i^2 - 2\,\widehat{FRR}\sum_{i=1}^{n}(a_i g_i) + \widehat{FRR}^2 \sum_{i=1}^{n} g_i^2}{\bar{g}^2 n(n-1)}, \end{aligned} \tag{4.29}$$

where a_i denotes the number of FR for client i, $\widehat{FRR}_i = a_i/g_i$ denotes the estimated FRR for speaker i and $\bar{g} = (1/n)\sum_{i=1}^{n} g_i$ is the average number of trials per client.

In case that the number of trials is equal for all clients ($g = g_i \;\forall\; i;\; \bar{g} = g$) the expression simplifies to

$$\hat{V}(\widehat{FRR}) = \frac{\sum_{i=1}^{n}(\widehat{FRR}_i - \widehat{FRR})^2}{n(n-1)} = \frac{1}{n-1}\left(\frac{\sum_{i=1}^{n} a_i^2}{g^2 n} - \widehat{FRR}^2\right). \tag{4.30}$$

When each client supplies only one trial, we receive the estimated variance of $\widehat{FRR} = \frac{1}{n}\sum_{i=1}^{n} a_i$:

$$\hat{V}(\widehat{FRR}) = \frac{\widehat{FRR}\,(1 - \widehat{FRR})}{n(n-1)}, \tag{4.31}$$

which is the well-known estimated variance of an estimator of the proportion in a Bernoulli experiment.

FAR

The situation turns out to be more complex when trying to estimate the variance of the FAR. Wayman [40] cites a formula provided by Bickel for the rather limited case when a full set of cross comparisons is made and one sample per speaker for testing is provided. Each of the n clients in the data set acts as impostor for the remaining $n-1$ clients. In

[5], section 6.3 the formula is extended for an arbitrary number h of trials per impostor. We require the following additional definitions:

n	number of client models
h	number of trials per impostor
b_{ji}	number of FA when impostor j violates model i
$c_i = \sum_{j=1}^{m} b_{ji}$	total number of FA against model i
$d_j = \sum_{i=1}^{n} b_{ji}$	total number of FA made by impostor j
$\widehat{FAR} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{m} b_{ji}}{hn(n-1)}$	estimated FAR for the cross comparison case.

The variance of the estimated FAR rate is given by

$$\begin{aligned}\hat{V}(\widehat{FAR}) &= \frac{\sum_{i=1}^{n}(c_i+d_i)^2 - \sum_{i=1}^{n}\sum_{\substack{j=1\\ j\neq i}}^{m}(b_{ji}^2 + b_{ji}b_{ij})}{h^2 n(n-1)(n-2)(n-3)} - \frac{4n-6}{(n-2)(n-3)}\widehat{FAR}^2 \\ &\approx \frac{1}{h^2 n^2 (n-1)^2}\sum_{i=1}^{n}(c_i+d_i)^2 - \frac{4}{n}\widehat{FAR}^2. \end{aligned} \tag{4.32}$$

We will refer to the variance estimations for FRR and FAR of this section (4.29 and 4.32) as "sub-sampling type 1" later on.

Estimating confidence intervals

With sufficiently large samples, the central limit theorem applies and the estimated error rates will follow approximately the normal distribution. In the case of FRR e.g. the confidence intervals are estimated as

$$\left[\widehat{FRR} - z_{\alpha/2}\sqrt{\hat{V}(\widehat{FRR})} \quad , \quad \widehat{FRR} + z_{\alpha/2}\sqrt{\hat{V}(\widehat{FRR})}\right]. \tag{4.33}$$

For the 95% confidence interval, the value $z_{\alpha/2} = 1.96$ with $\alpha = 0.05$.

Confidence bounds based on Beta-binomial distribution

The binomial approach for estimating the confidence intervals assume a constant error rate over the population. However, several authors [8, 45] report varying error rates within parts of the population and the already mentioned term "Doddington's zoo" divides the population into groups according to their contribution to decision errors.

To complete the overview of different approaches for variance estimation, we present briefly an approach developed by Schuckers [46, 47], the so-called "Beta-binomial" distribution. It uses the Beta distribution for modelling the variability in the error probability among individuals:

$$f(p_i|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p_i^{\alpha-1}(1-p_i)^{\beta-1} \tag{4.34}$$

Conditional on the parameters α and β, the distribution is J-shaped or reverse J-shaped when $\alpha > 1$ and $\beta > 1$. In the case of the sum of both parameters approaching large values, the Beta-binomial distribution approaches the binomial model. Estimating the parameters using maximum likelihood (ML) requires numerical computation as no closed form solution exists.

Beside the computational effort for the Beta-binomial distribution, its greatest disadvantage is

that it is inappropriate for bimodal error distributions caused by "goats" and "sheep". Schuckers suggests a mixture of Beta distributions for future work.

Since the application of the Beta-binomial distribution requires significantly more computational effort, we will not include this approach in our further comparisons.

4.4 Empirical confidence bounds using Bootstrap estimates

Bootstrapping is a standard statistical technique that was originally developed by Efron in 1979 for calculating confidence intervals where standard methods can not be applied (see [48] for a good introduction). It is related to other computer-intensive statistical methods such as subsampling or the jackknife. The main idea of the bootstrap is to generate more data samples from an existing one by randomly reassigning observations with replacements. The term "bootstrap" comes from the analogy with the famous story "Baron Münchhausen". Spoken in simple words, the bootstrap method does artificially what an experimenter would do in practice: repeat an experiment several times.

The Bootstrap principle

Suppose a parameter $\theta(F)$ should be estimated for a population with distribution F of unknown form and $\boldsymbol{X}$ is an i.i.d. sample from this population. As F is not known we can only estimate the parameter by a statistic $T = T(\boldsymbol{X})$ based on the sample. If the population and its distribution F *were* known, we would be able to calculate (analytically or by Monte Carlo simulation) key parameters of the estimate T e.g. $\mathrm{Bias}_F(T)$ or $\mathrm{Var}_F(T)$. The bootstrap method uses the simple idea: since you do not have the whole population, use the observed sample $\boldsymbol{X}$ and use it for Monte Carlo simulations. Therefore the bootstrap paradigm is based on the assumption that the observed sample $\boldsymbol{X}$ exactly represents the whole population with its distribution F. The observed sample $\boldsymbol{X}$ itself consists of N observations $\{X_1, \ldots, X_N\}$.

The bootstrap procedure for estimating confidence intervals for the parameter $\theta(F)$ includes the following steps:

- Draw B i.i.d. samples $\boldsymbol{X}^{*(1)}, \ldots, \boldsymbol{X}^{*(B)}$, each of size N from the sample population consisting of the observations $\{X_1, \ldots, X_N\}$. These B i.i.d. samples are called *resamples* in bootstrap terminology. Drawing observations from $\boldsymbol{X}$ is done with replacement.
- Using the B resamples, calculate B corresponding estimates of the parameter, denoted by $T^{*(i)}$.
- The distribution function of the parameter $T(X)$ is $G_{T,F}(x) = P_F(\, T(X) \leq x \,)$.
- The bootstrap approximation of $G_{T,F}(x)$ is denoted as $G^*_{T,F}(x)$, its $\alpha/2$ and $(1-\alpha/2)$ quantiles are $q^*(\alpha/2)$ and $q^*(1-\alpha/2)$, respectively. The quantiles can be computed by determining the value x for which the percentage q^* of the B sorted estimates T^* are smaller than x.
- Finally, the $(1-\alpha)$ bootstrap confidence interval for $\theta(F)$ would be

$$[q^*(\alpha/2)\,,\, q^*(1-\alpha/2)]\,. \tag{4.35}$$

The bootstrap is valid for i.i.d. samples, which occur in many cases where the number of observations is high and the observations are independently distributed. Please refer to [48], [49] and [50] for an in-depth description of the bootstrap and its applications.

Bootstrapping for estimating confidence bounds of DET curves

There are various approaches in applying the bootstrap technique for the estimation of confidence bounds of the presented performance figures FRR, FAR and the DET curves. Each method differs in the generation of the resamples.

Common to all techniques is the procedure to create the confidence bounds: all DET curves received by the bootstrap resampling described above are interpolated to a common subset of thresholds. For each of these thresholds, the set of FRR and FAR are sorted separately and the confidence intervals for the error rates are stored.

The threshold resampling using interpolation ensures that corresponding points of the set of DET curves get selected.

In the following we will present three different bootstrapping techniques which differ mainly in the way how the bootstrap resamples are drawn from the observed sample $\boldsymbol{X}$:

Bootstrapping type 1
Bolle et. al [51] proposes drawing bootstrap samples from the set of matching scores (genuine tests) and from the non-matching scores (impostor tests) to construct confidence intervals for FRR and FAR, respectively. Evaluating the interval for any decision threshold will give the confidence band of the DET curve when plotting the horizontal confidence interval for each FAR and the vertical confidence interval at each FRR. Naturally as the number of client trials is often inherently lower than the number of impostor trials, the confidence regions of FRR are wider. We will refer to this type of bootstrapping later in the text as "bootstrapping type 1".

Bootstrapping type 2
A more refined resampling scheme for the variability of the FAR is presented in [5]. The impostor scores are collected using the cross comparison scheme mentioned on page 31. Let $S(v, t, c)$ be the result of a cross comparison of impostor v with its trial t against client c. A bootstrap sample is constructed from $\boldsymbol{S}$, the full set of cross-comparisons, in a way that replicates the original sample structure and dependencies:

- Sample k impostors with replacement: $v(1), \ldots, v(k)$.
- For each $v(i)$: sample with replacement from the $(n-1)$ non-self clients: $c(i,1), \ldots, c(i,n-1)$.
- For each $v(i)$: sample with replacements h trials made by that impostor: $t(i,1), \ldots, t(i,h)$.
- Finally the bootstrap sample can be constructed:

$$\boldsymbol{S}^* = \{\, S(v(i), t(i,k), c(i,j)) \mid i \in \{1,\ldots,k\},\ j \in \{1,\ldots,(n-1)\},\ k \in \{1,\ldots,h\} \,\}.$$

A similar sampling scheme can be constructed analogously for the client scores by skipping the second step in the former scheme. This more refined bootstrapping scheme will be denoted as "bootstrapping type 2".

Resampling for cross-comparisons
A small modification of the previous scheme is applied by Wayman [40]. In order to compare the confidence bands received via bootstrapping with the confidence bands predicted by the subsampling assumption presented above, the scores of the subsample have to comply with the cross-comparison criterion. Wayman draws therefore the clients without replacement

and constructs the impostor population from this client set. Drawing with replacement would otherwise allow the possibility that a single client could be compared to himself as impostor. The number of trials is fixed to one in his approach. We will denote this type as "resampling cross-comparisons" since this kind of resampling scheme does not follow the bootstrap resampling principle.

The two variations "Bootstrapping type 1" and "Bootstrapping type 2" operate on scores from a fixed population of trained client models. Therefore both methods can only capture the variation within the underlying allocation of individuals to different sets according to the evaluation scheme. Using a cross-comparison scheme renders the score population independent from the allocation between client and test set.

However, in case of speaker verification with score normalisation using a world model, the allocation of speakers to this third set becomes an influencing factor. Also the selection of the training material for the client models can show a high impact. A more advanced bootstrap sampling scheme could create bootstrap samples for both training and testing data. A verification system is trained using the bootstrap training sample and the evaluation is performed using the bootstrap test set. This procedure is repeated for all bootstrap samples.

Macskassy et al. [52] propose such an approach for a classifier using a Probability Estimation Tree. This type, is clearly more computational intensive as the techniques working only with a fixed score distribution. However, it captures the variation which is introduced by partitioning the data into a training and a test set.

4.5 Practical construction of confidence bounds

Construction of DET plot confidence bounds

Confidence bounds allow to judge the statistical relevance of the results of two different SV systems labeled A and B. The confidence of performance differences can be judged by testing if the DET curve of system B lies within the confidence bounds of the DET curve of system A. However, the concept of the notion "lie within" can be regarded in different ways.

Several possibilities exist for constructing the confidence bands for the DET plots. As there are a horizontal interval (for FAR) and a vertical interval (for FRR) for each point of the curve, it is not obvious how to construct a two-dimensional confidence area from two one-dimensional intervals. We decided to use a worst-case band where the lower bound is defined by the lower bound of FRR and the lower bound of FAR (as defined by the respective confidence intervals). The underlying assumption is that there might be an unknown interaction between both variances. In case of totally independent variations in the estimated error rates, this technique will overestimate the true confidence band and thus give a rather pessimistic picture. However, as all confidence bounds are constructed in the same way, comparisons among different approaches are consistent.

The upper bound is defined similar to above by using the upper bounds of FRR and FAR. Figure 4.4 depicts the construction of the confidence bounds for a single point on the DET curve.

In [52] some of the possible confidence band construction techniques are presented and investigated with regard to the containment of the true DET curve. Unfortunately, the method used here is not included in the comparison.

We will later use the construction scheme that we described above for plotting empirical bounds as well as predicted bounds.

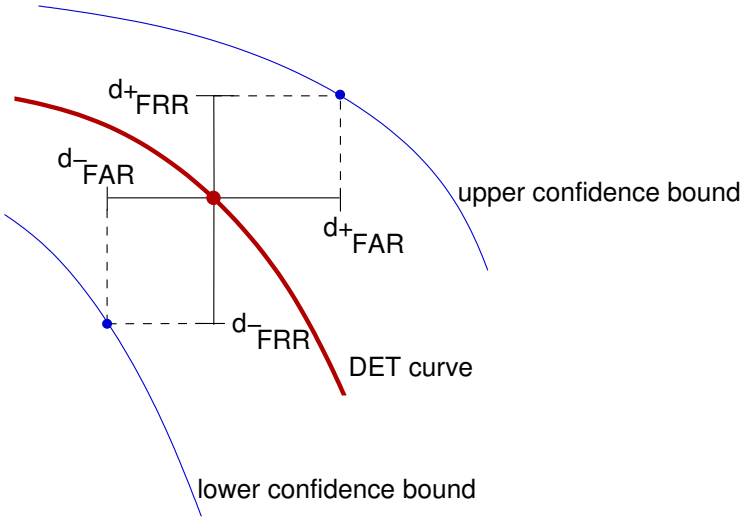

Figure 4.4: Construction of the confidence bounds for a single point of the DET curve. $d-_{F..}$ and $d+_{F..}$ denote the lower and upper half of the confidence interval of the respective error rate.

Construction of the EER confidence bounds

In practical work it is desirable to receive a confidence interval for the EER in order to compare different systems. With some assumptions this confidence interval can be derived from the confidence intervals of both FR and FA that were already used in the previous section to construct the confidence bounds of the DET curve.

Figure 4.5 shows a magnified section from a DET plot with linear scales on both axis. We assume here that we can approximate the DET curve using linear sections and that further the confidence intervals are so small that we do not leave the area where our assumption of linearity is valid. The direction of the DET curve is given by the normalised direction vector $\boldsymbol{det}$. In our example we plot only the upper confidence bound that is described by the same direction vector $\boldsymbol{det}$. The EER is determined by the intersection of the DET curve and the diagonal line given by the normalised direction vector $\boldsymbol{w} = \frac{\sqrt{2}}{2}\binom{1}{1}$. The upper confidence bound of the EER, denoted as EER_u, is determined by the upper confidence bound of the DET. The vector $\boldsymbol{e}$ denotes the difference between both EER points; its components describes the difference between both EERs ($d+_{EER} = \mathrm{EER}_u - \mathrm{EER}$).

A simple geometrical calculation gives the upper half of the EER confidence bound as

$$d+_{EER} = \left| \frac{det_{FAR}\, d+_{FAR} \;-\; det_{FRR}\, d+_{FRR}}{det_{FAR} \;-\; det_{FRR}} \right|, \tag{4.36}$$

where det_{FAR} denotes the component of the direction vector $\boldsymbol{det}$ in the direction of the FAR-axis (x-axis) and det_{FRR} denotes the corresponding component in the direction of the FRR-axis (y-axis).

In the common case that the DET curve is orthogonal with respect to the diagonal $\boldsymbol{w}$, the calculation simplifies to

$$d+_{EER} = \frac{d+_{FAR} \;+\; d+_{FRR}}{2} \tag{4.37}$$

We assume that the lower confidence bound of the EER can be constructed symmetrically

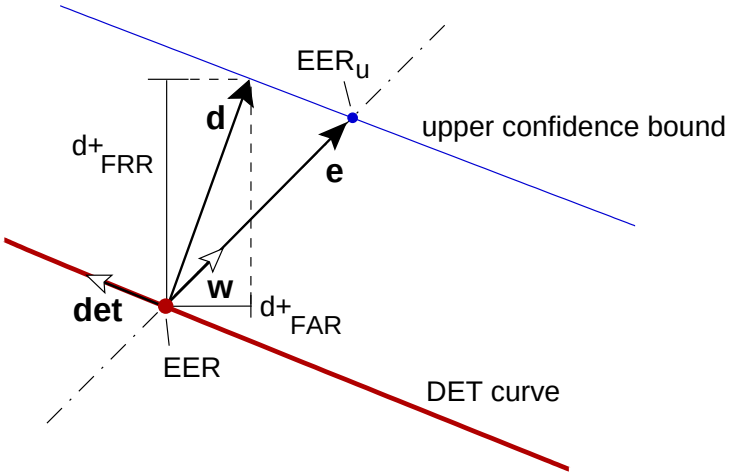

Figure 4.5: Calculation of the confidence bounds of the EER from the confidence bounds of the DET curve (displayed in a linear scale). ***det*** denotes the normalised direction vector of the DET curve, $\boldsymbol{w}$ denotes the normalised direction vector of the diagonal. The vector $\boldsymbol{d}$ is composed by the upper halves of the confidence intervals of both the FA and FR (denoted as $d{+}_{FAR}$ and $d{+}_{FRR}$ in figure 4.4). The upper half of the EER confidence bound ($d{+}_{EER}$) can be derived from either component of the vector $\boldsymbol{e}$ which in turn is determined by the vectors ***det***, $\boldsymbol{w}$ and $\boldsymbol{e}$.

($d{+}_{EER} = d{-}_{EER}$) which finally gives the confidence interval of the EER:

$$\mathrm{EER} \in [\mathrm{EER} - d{+}_{EER} \quad \mathrm{EER} + d{+}_{EER}] \,. \tag{4.38}$$

In our following investigations we will deploy confidence intervals for EER for $p = 0.05$ because the corresponding confidence intervals of FAR and FRR are also calculated for this error level. In addition, it turned out that all DET plots on a linear scale run diagonal in the area of the EER. Thus we use the simplified formula 4.37 for the EER confidence interval.

4.6 Training and evaluation schemes

Having presented common performance figures and the calculation of their confidence bounds, we need to partition the data of the VeriDat corpus for a meaningful evaluation in the domain of speakers as well as in the domain of recorded items. First, we describe the speaker sets used for our evaluation scheme. Following that, we focus on the training and the two different evaluation session sets developed for the VeriDat database which are applied in the experiments in the following chapters.

Selection of speaker sets

A complete evaluation scheme of any kind of pattern recognition system normally requires three different kinds of data sets:

- A training set. Training of models or template generation requires a (in most cases: pre-labeled) data set. In case of a speaker verification system each speaker from the client set provides a set of utterances from which the model or template is generated.

- A tuning set. Some systems require adjustment of parameters done with an additional set. This additional speaker set can be accounted to the training set but should not overlap with the client population.
- A test set: A separate data set with material never seen during training is used to evaluate the performance of the models/templates. This means for a speaker verification system that for the client speaker set additional recordings for evaluating the genuine access must be available. For impostor tests, another speaker set (impostor speakers) has to be defined.

For methodically sound experiments all three data sets should be disjunct. Otherwise, unpredictable interaction can occur and performance figures can be biased. The sets defined above are somehow still ambiguous for a biometric system using a model for each registered client. This fact does imply a two-fold partitioning: one of the individuals involved in the evaluation and one of the trials supplied by each individual.

On the first level of the two-stage partitioning we defined the following four speaker sets from the total 150 VeriDat speakers:

- Client set (30 speakers). Individual models are build for these speakers.
- World set (30 speakers). Used for building the world model(s) (might also be used for other score normalisation methods e.g. cohort normalisation).
- Impostor set (60 speakers). Used for evaluation of the client models with non-genuine claims.
- Development set (30 speakers). Used for calculating various parameters (e.g. world model quality, a priori thresholds, etc.).

We will postpone the description of the second stage of the partitioning scheme (partitioning according to the recorded items) to the next subsection. Before that, the selection scheme for the first stage is presented in more detail.

In our selection scheme of speaker sets, the tuning set from the general evaluation scheme is split into two sets of equal size, the world set and the development set. Commonly used normalisation techniques require a set of speakers known as *background set* or *world set*. The development set itself can be regarded as a tuning set whenever setting various parameters of the world models.

A purely random selection of each of the 150 speakers to one of the four sets would certainly produce a non-representative distribution of speakers in the four sets. Due to the small size of the speaker population and the sets, we applied some constraints in order to achieve a representative distribution of the speakers in each set. Using the general information about the speakers, we generated the sets with a common distribution in gender, accent and age group. A tolerance value is required when comparing the distributions in order to determine a set partitioning from the nearly unlimited number of possible partitions. For the maximum absolute deviation of the property distributions from the targeted distributions, the value '3' was found experimentally; it is a tradeoff between reasonable computation duration and adequate match of the property distributions.

Since the VeriDat database also provides information about the family relationship among the speakers, we preferred the selection of related speakers for the client and the impostor set. If a speaker exhibits one or more relationships (e.g. father with two children) the rule is to assign the speaker to the client group and its relatives to the impostor group.

The selection algorithm summarised:

1. Randomly draw (without replacement) 30 speakers without relationship of the total population for the world set.
2. Test the world set with regard to adequate distribution of gender, accent and age group. Go back to 1.) if the constraints are not met.
3. Construct the development set with 30 speakers alike 1.): draw from the remaining population having no relationship.
4. Test the development set with regard to adequate distribution of gender, accent and age group. Go back to 1.) if the constraints are not met.
5. Randomly draw 30 speakers of the remaining population and assign them to the client set.
6. Test the client set with regard to adequate distribution of gender, accent and age group. Go back to 1.) if the constraints are not met.
7. All clients with relationships must belong to distinctive relation groups. If not, go back to 1.).
8. Assign the remaining speakers (60) to the impostor set.
9. Test the impostor set with regard to adequate distribution of gender, accent and age group. Go back to 1.) if the constraints are not met.
10. Store the final set and finish.

The final set includes 13 male and 17 female speakers in the client set and 30 male and 30 female speakers in the impostor set. A more detailed compilation of the speakers, their properties and their set allocation is provided in the appendix B.

Selection of training sets

The second stage of the data partitioning handles selection of training and evaluation trials for each client in the system. In case of the VeriDat database the natural selection unit would be a set of items belonging to a single recording session.

Training set "A-Base tr" and "AW-Base tr"

We have stated earlier in chapter 4.1 that several factors influence the selection and amount of enrolment material. Practical constraints demand that a speaker verification system uses a minimum of enrolment material. When using the number triplets described earlier on page 145 we selected 28 triplets (equivalent to data from four sessions) for enrolment. For our experiments we decided that this amount could pose an acceptable effort for potential clients.

A natural selection scheme of training sessions would be the chronological order in which the recordings were made. Unfortunately, the distribution of the four different recording conditions is unbalanced within the first few sessions of VeriDat: the first session of type "Fixed/Noisy" is foreseen as seventh session in the recording scheme. We decided to select always the first session of each recording condition leading to the session set: 01 ("Fixed/Quiet"), 02 ("GSM/Quiet"), 04 ("GSM/Noisy") and 07 ("Fixed/Noisy"). This training set is denoted as "A-Base tr".

Preliminary experiments [53] showed that the best performing speaker models were those that were trained with data selected from all possible recording conditions. Thus we defined the training set "A-Base tr" which uses all possible session variations in the data.

The corresponding training set for the world models which is denoted as "AW-Base tr" contains all 140 triplet items (20 sessions).

Training session sets "B" and "BW"

The investigations made in [53] require also special training material selected from only one subcorpus of VeriDat (either "Fixed" "GSM", "Quiet", "Noisy") or any combination of these. The total number of items in the training set is reduced to 20 as for each speaker the recordings with type "GSM/Noisy" and "Fixed/Noisy" contain only 42 items. Six different "B" schemes are defined with items selected from:

B-Base all recording sessions;

B-FQ only the "Fixed/Quiet" sessions;

B-F the "Fixed/Quiet" and the "Fixed/Noisy" sessions;

B-Q the "Fixed/Quiet" and the "GSM/Quiet" sessions;

B-G the "GSM/Quiet" and the "GSM/Noisy" sessions;

B-N the "Fixed/Noisy" and the "GSM/Noisy" sessions.

Since these training sets were originally intended also for use with sub-word models, a balanced occurrence of all possible number sub-words was required. In appendix A.3 we discuss the problem that the sub-words are not equally distributed within a session. Therefore we decided to compose the training sets from various sessions. The motivation for this type of training item selections are the experiments presented in chapter 9, that require the same amount of training data for fair performance comparison.

The same training sets are defined for training of the world models using 42 training utterances. Because these models are tested using the development speaker set, we can deploy all available items from these speaker and do not need to keep some items for the evaluation set. The upper limit of the number of training items is defined by both noisy subsets, "GSM/Noisy" and "Fixed/Noisy", having each 42 items. The training sets are denoted with the prefix "BW-" instead of "B-". Similar to the explanation in the previous section, the comparison of equally trained models require this careful item selection.

Detailed listings of the "B-" and "BW-" training and evaluation sets are provided in appendix E.

Selection of evaluation session sets

Recordings of a client speaker used for enrolment were not deployed for testing. E.g. the training set "A-Base tr" is associated with the evaluation set "A-Base ev" which contains 112 items (140 utterances minus 28 training utterances). Only the genuine tests (scoring client recordings with the matching model) are done with this evaluation set. Impostor test (scoring speaker recordings with non-matching model) use all 140 triplet items of a single non-genuine speaker. For the "BW-" evaluation sets, we refer again to the detailed listing in appendix E.

The combination of the training sets with their matching evaluation sets is denoted here as the *item scheme*. E.g. the training set "A-Base tr" for the client models together with the world training set "AW-Base tr" and the matching evaluation set "A-Base ev" is denoted shortly as the item scheme "A-Base".

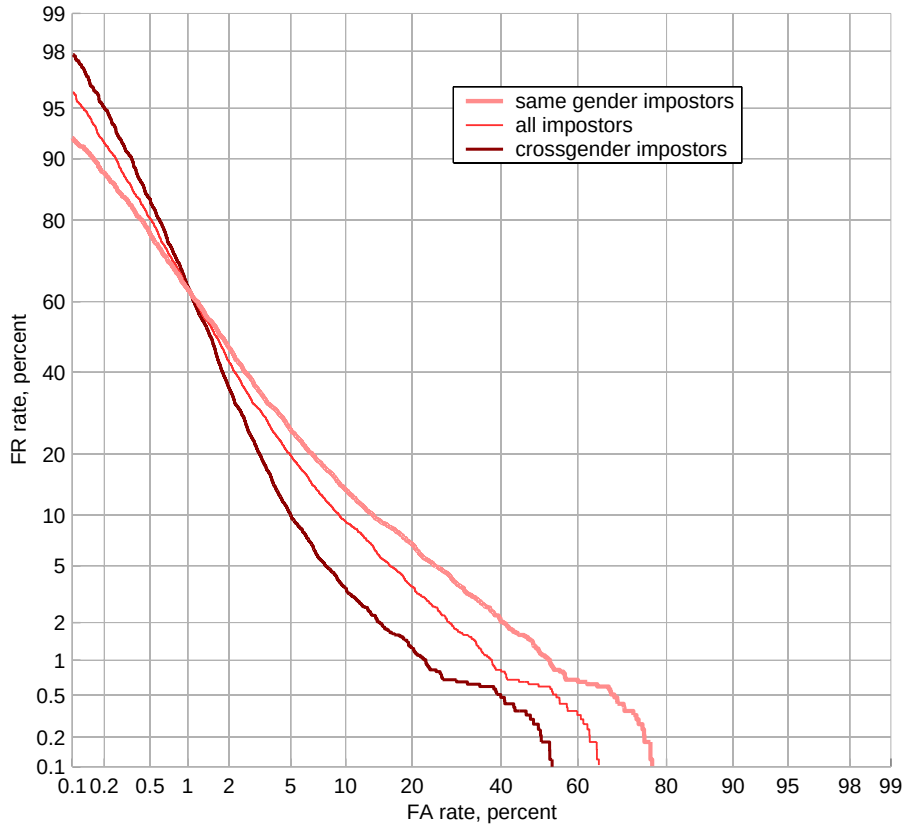

Figure 4.6: Example of DET plots with different impostor subsets. GMM SV system (four mixtures, feature set MFCC_SD, training set "A-Base tr", test protocol "separate impostors").

Selection of impostor subsets

In order to make the selection of impostor speakers more realistic, there is the option to compare impostors only to clients of the same gender. For the experiments described here, same-gender tests are reported.

Figure 4.6 shows the DET curves of a standard speaker verification system with different impostor selection paradigms. Near the area of the EER the performance increases as expected when going from same gender impostors to the complete impostor set and finally to cross-gender impostors. Naturally, when assuming unintentional impostors, the same-gender impostor subset over-estimates the true FAR. As we can see, there remain several successful acceptances of cross-sex impostors. It is not automatically guaranteed that same-gender impostors perform always better over the whole set of operating points. In the low-FAR part, the situation reverses when cross-gender impostors achieve higher FARs at a fixed FRR. In section 4.2.4 we have already presented the relation between the slope of the DET plot and the ratio of the spread of both the genuine and impostor scores. The steeper slope of the DET plot in this area is related to a wider spread of scores of the cross-gender impostor tests compared to the tests done with all impostors or the same-gender impostors. Only the composition of the impostor population changes, the distribution of the genuine scores is fixed. A wider distribution of the impostor scores reaches farther into the distribution of genuine scores which results in a higher FRR for the same FAR.

test type	client tests	impostor tests
all imps.	**3,360** (= 30*112)	**252,000** (= 30*60*140)
same gender imps.	**3,360**	**126,00** (= 13*30*140 + 17*30*140)
cross-gender imps.	**3,360**	**126,000** (= 13*30*140 + 17*30*140)

Table 4.1: Test sizes for client tests and impostor tests for the "separate impostors" protocol. Training set "A-Base tr". Clients include 13 male and 17 female speakers, while impostors have each 30 male and female speakers.

Selected evaluation protocols

With regard to the aspects mentioned above, there is a need for two different protocols defining how impostor speakers are assigned to client speakers. Accordingly we created two different protocols for our experiments: "separate impostors" and "cross validation impostors". The later is only used for cross validation experiments in order to judge the validity of confidence bounds discussed in section 5.3.

We will report test sizes in the case of using the "A-Base tr" training set, which provides 112 utterances for the genuine test. Test sizes for the "B" training sets are calculated similarly.

Separate impostors

All four speaker sets selected by the method described on page 39 were utilised. Since there is no intersection between clients and impostors, we denote this protocol as "separate impostors". Depending on a possible subselection of impostors based on the client's gender, we have either 252,000 impostor tests using all impostors or 126,000 impostor tests in the same gender or cross-gender case (see table 4.1). In total there are 3,360 client tests.

Cross validation impostors

Since the formula of Bickel and Mansfield for estimating the confidence bounds of the false acceptance rate (4.32) requires data from cross validation experiments, we set up a second evaluation protocol with two variations.

The first version uses the same number of client speakers as in the "separate impostors" case. Due to the cross validation scheme, we use the remaining 29 speakers of the client set as impostors for a selected speaker. This version of the protocol is denoted as "cross-validation 30".

The second version merges the distinctive impostor set and development set into the client set and creates thus a larger set consisting of 120 client speakers. It is denoted as "cross-validation 120".

Table 4.2 summarises the test sizes of both protocols with regard to the variation in impostor selection.

test type	client tests	impostor tests
cross-validation 30		
all imps.	**3,360** (= 30*112)	**121,800** (= 30*29*140)
same gender imps.	**3,360**	**59,920** (= 13*12*140 + 17*16*140)
cross-gender imps.	**3,360**	**61,880** (=13*17*140 + 17*13*140)
cross-validation 120		
all imps.	**13,660** (= 120*119*112)	**1,999,200** (= 120*119*140)
same gender imps.	**13,660**	**993,720** (= 57*56*140 + 63*62*140)
cross-gender imps.	**13,660**	**1,005,480** (= 57*63*140 + 63*57*140)

Table 4.2: Test sizes for client tests and impostor tests for the two defined cross-comparison evaluation protocols. Training set "A-Base tr". Clients include 13 male and 17 female speakers, while impostors have each 30 male and female speakers.

Chapter 5

Evaluation of confidence estimation techniques

In the previous chapter we have already given a summary of commonly deployed estimation techniques for the false rejection rate (FRR), the false acceptance rate (FAR), the equal error rate (ERR) and the detection error trade-off plots (DET plots). Among others, we have presented on page 31 a statistical model for the score variability in a biometric systems which was introduced in [5] and [40].

The statistical model for calculating the confidence intervals of the FRR bases on a well known model for two-stage sampling design (see e.g. [44, section 21.5]). The confidence intervals of the FAR are given by an unreconstructable formula (4.32) provided by Bickel in [40] for the case that a cross comparison scheme for selecting the impostors is used.

Our aim in this chapter is to develop an estimation formula for the FAR confidence intervals similar to the FRR confidence interval formula 4.29.

First, we will show that a single requirement of the binomial approach for the confidence bounds (page 29), namely the assumption of a global error rate, is not valid with real data. Since there seems to be speaker effects in the scores we extend the two-stage sampling model of the subsampling type 1 (see page 31) that was used for the genuine trials, and postulate a three-stage sampling model for the non-genuine trials. This new approach will be named "subsampling type 2".

We will compare the resulting confidence bounds with Bickel's approach (denoted as "subsampling type 1") and the binomial bounds that we have already explained on page 29. In anticipation of the experiments later on we will use some of the performance results obtained there and compare experimentally obtained confidence bounds with the predicted bounds.

For the measured confidence bounds, we will deploy the already outlined bootstrap technique from section 4.4 by using a speaker verification system using GMMs with four mixtures and the triplet data for training and evaluation.

A discussion of selected confidence bound estimates and their impact on comparing experimental results of the following chapters will conclude this chapter.

5.1 Validity of the binomial assumption

In [8] the score distributions of several different speaker verification systems have been analyzed in order to find evidence for the existence of speaker groups, in particular 'goats', 'lambs' and

'wolves'. Although the authors could not clearly divide the participating speakers into distinct classes, they proved that speaker effects exist and that the contribution to the error rates FAR and FRR are not equally distributed within the speaker population.

We conducted similar tests with the baseline system that we use in this section: the GMM SV system with four mixtures, feature set MFCC_SD, training and test protocol "separate impostors".

A first analysis concentrated on the distribution of the FRR over the speaker population. Here we deployed the test protocol "cross-validation 120". Figure 5.1 shows the histogram of the FRR of 120 speakers. The phenomenon of a speaker population consisting of "sheep"-like and "goat"-like speakers, also known as "Doddington's zoo", has already been mentioned in section 2.4. Typically, it can not only be found for speech but also for other biometrical modalities as well.

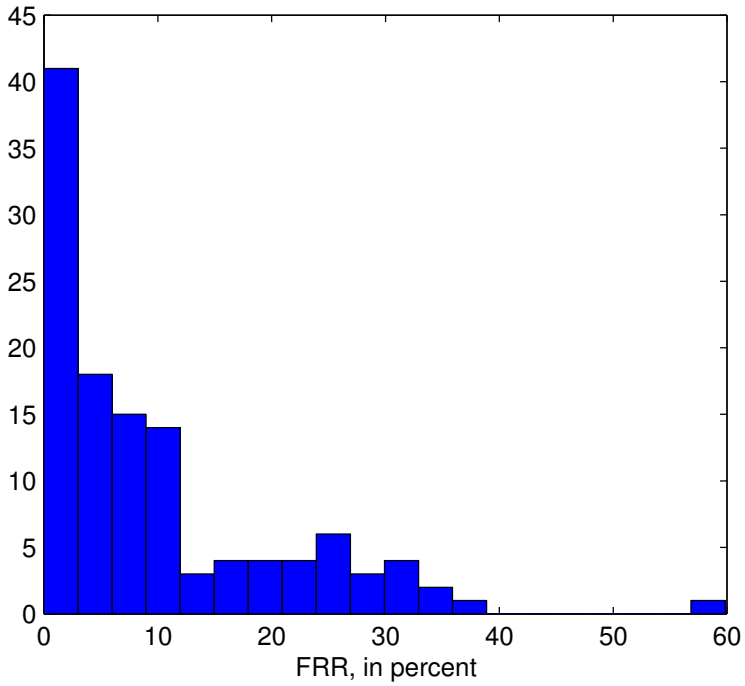

Figure 5.1: Histogram of the FR rates of 120 speakers at the EER operating point. GMM SV system (four mixtures, feature set MFCC_SD, training and test protocol "cross-validation 120").

Since the FR rates are based on the underlying scores, we conducted a further analysis on the score distribution for each speaker. In order to minimize the environmental effects, we selected only the recordings from the "Fixed-Quiet" set. Again, we concentrate on the scores from the genuine tests which involve the 30 speakers from our standard client speaker set.

The calculation of the variability of the scores from each speaker showed that only the speakers '0011', '0044', '0047', '0059', '0097' and '0119' have a nearly equal score variability. This is the basic requirement for the Kruskal-Wallis test, a non-parametric one-way analysis of variance by ranks test. The hypothesis, that there is no speaker effect in the score population could be easily rejected at the 0.01 significance level ($\chi^2 = 71;\ p \approx 0$).

Similar results can be obtained for the scores of the impostor speaker tests. Here we find the existence of two effects: the effect of the impostor speaker and the effect of the claimed speaker model.

Thus, we argue that the inherent structure of the scores must be regarded and the bootstrap sampling can only assume an i.i.d. population of scores within the data of an individual speaker. These constraints are met by the bootstrap sampling type 2 that respects the information about the speakers and their trials when drawing bootstrap samples. For the impostor speaker tests, the identity of the speaker model is further regarded.

5.2 A general sub-sampling scheme (subsampling type 2)

In the last section we have already listed evidence for the assumption that the binomial approach generally underestimates the confidence bounds since the total score population can not be sufficiently described by a global error rate. A different confidence estimation scheme has been presented on page 31, the 'subsampling type 1' scheme. It is based on an extended statistical model, also known in statistics as the *random-effects model*. However, the estimated variance of the FAR is limited to the cross-validation case.

In addition we found that the impostor speaker scores show similar speaker effects compared to the genuine speaker scores.

We propose a modified estimation scheme for the variance of the FAR that is based on the same statistical model used for estimating the variance of the FRR (4.29).

While this model used with two stages describes the situation for the genuine tests, its extension to three stages seems to be suited for the impostor tests. Both cases are described more detailed in the following.

Two-stage sampling

The application of the random effects model (see e.g. [44], chapter 13.1) assumes that the genuine score S_i^k of speaker i with utterance k can be written as

$$S_i^k = \mu + C_i + \epsilon_{ik}. \tag{5.1}$$

The symbol μ denotes the mean score of the total population of scores, C_i represents the difference between the scores of speaker i and the total mean μ. It is assumed that this variation can be described with a normally distributed C_i with variance σ_C^2, hence $C_i \sim \mathcal{N}(0, \sigma_C^2)$. The term ϵ_{ik} captures the variation within the scores of a single client and is itself normally distributed, $\epsilon_{ik} \sim \mathcal{N}(0, \sigma^2)$.

This type of model can be used to describe the variance of parameters estimated with two-stage sampling. On each stage, samples are drawn from a new subset. The first stage corresponds to the clients that are drawn virtually from an unlimited population. On the second stage we draw from the selected client's population of achievable scores.

Estimation of the proportion of FR among the trials is given by formula 4.9. The variance of this estimate is

$$\hat{V}(\widehat{FRR}) = (1 - f_1)\frac{s_1^2}{n} + f_1(1 - f_2)\frac{s_2^2}{n(g-1)}, \tag{5.2}$$

with

$$s_1^2 = \frac{\sum_{i=1}^{n}(\widehat{FRR}_i - \widehat{FRR})^2}{(n-1)} \tag{5.3}$$

$$s_2^2 = \frac{\sum_{i=1}^{n}\widehat{FRR}_i(1-\widehat{FRR}_i)}{n(g-1)}. \tag{5.4}$$

Again we assume that each client provides g trials.

The terms f_1 and f_2 denote the finite population correction. They enter when we are sampling from a population of finite size instead from an infinite population. In the latter case, they can be set to zero. We use this approximation as we do not know the total speaker population nor the total population of utterances. This simplifies the variance to the form already given in 4.29:

$$\hat{V}(\widehat{FRR}) = \frac{\sum_{i=1}^{n}((\widehat{FRR}_i - \widehat{FRR})^2}{n(n-1)}. \tag{5.5}$$

Three-stage sampling

The same random effects model can be extended to three stages which will comprise sampling from the impostor population on the first stage, sampling from the client models on the second stage and finally sampling from the trials on the third stage.

$$S_{ij}^k = \mu + D_j + C_{ji} + \epsilon_{jik}. \tag{5.6}$$

The symbol μ denotes the mean score of the total population of scores, D_j denotes the difference between the scores of impostor j and the total mean μ, C_{ji} represents the difference between the scores of the client model i and the mean score of impostor j. The term ϵ_{jik} captures the variation within the scores of a single impostor against a single client model. Following the same scheme as in the two-stage sampling case we can estimate the variance of FAR as

$$\hat{V}(\widehat{FAR}) = \frac{(1-f_1)}{m}s_1^2 + \frac{f_1(1-f_2)}{mn}s_2^2 + \frac{f_1 f_2(1-f_3)}{mnk}s_3^2. \tag{5.7}$$

The three variances s_1^2, s_2^2 and s_3^2 are defined as

$$s_1^2 = \frac{\sum_{j=1}^{m}(\widehat{FAR}_j - \widehat{FAR})^2}{(m-1)}, \quad \text{variance between impostors,} \tag{5.8}$$

$$s_2^2 = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n}(\widehat{FAR}_{ji} - \widehat{FAR}_i)^2}{m(n-1)}, \quad \text{variance between client models, within impostor,} \tag{5.9}$$

$$s_2^2 = \frac{\sum_{j=1}^{m}\sum_{i=1}^{n}\widehat{FAR}_{ji}(1-\widehat{FAR}_{ji})}{mn(h-1)}, \quad \text{variance between utterances, within client model.} \tag{5.10}$$

We use the same argument as above for setting the finite population corrections f_1, f_2 and f_3 to zero. Finally we receive the estimated variance of the estimated FAR

$$\hat{V}(\widehat{FAR}) = \frac{\sum_{j=1}^{m}((\widehat{FAR}_j - \widehat{FAR})^2}{m(m-1)}. \tag{5.11}$$

The confidence intervals are again estimated using the normal distribution approximation.

In order to distinguish the two different approaches when estimating the variance of FAR, we will denote our proposed estimators as "subsampling type 2".

5.3 Comparison of empirical bounds with predicted bounds

In section 4.3 we presented several approaches to estimate the variance and confidence bounds of the performance parameters FRR and FAR. Some of these techniques for predicted bounds, namely the binomial assumption and its approximation by the rule of 30 (page 29), the sub-sampling type 1 (page 31) and its modification, the sub-sampling type 2 (see previous section) are applied to experimental performance measures in this section.

These predicted bounds will be compared with empirical bounds estimated by bootstrap techniques presented in section 4.4, namely the "bootstrapping type 1", "bootstrapping type 2" and "resampling for cross-comparison". Each of these bootstrap techniques are closely related to one of the theoretically founded models for estimating the confidence bounds.

Our final goal is to select a single variance estimation technique which suits best for our following experiments.

All reported confidence bands are constructed from the symmetrical 95% confidence bounds. The confidence bounds generated via bootstrap are all based on 200 bootstrap samples.

We will make extensive use of DET plots in this section because they provide easy presentation of both confidence bounds (FRR and FAR) in a single view.

5.3.1 Comparison of binomial bounds and bootstrapping type 1 and 2

Throughout this subsection we will apply the "separate impostors" test protocol. Only the binomial error bounds are generated and compared to a bootstrap estimate of the confidence band using bootstrap sampling type 1 and type 2.

Preliminary experiments showed, that the bootstrap sampling type 1 matches nearly exactly with the predicted binomial bands. Both techniques are based on the same assumption: the total population of the scores is captured as a whole entity, without any further structure within the scores. In terms of the binomial assumption, it is described by a global error rate while the bootstrap sample drawing assumes an i.i.d. score population that is represented by the observed sample $\boldsymbol{X}$.

Figure 5.2 shows clearly that the binomial confidence bounds lie very tight around the DET curve. The approximation of the exact binomial bounds made by the Rule of 30 is valid over a wide range of the DET curve i.e. both curves lie on top of each other. Only in the high FRR part, the factor f in formula 4.23 is overestimated by the rule of 30 and hence the confidence band widens.

Despite the fact that the binomial bounds are not considered as valid for real data, the comparison using bootstrapping type 1 was not done in vain as it shows the validity of the bootstrap approach for our experiments in general.

Again from figure 5.2 we can see that the confidence bands based on bootstrapping type 2 are much wider than the binomial bounds and clearly show that the binomial bounds are far more optimistic about the true variance of the DET curve than the empirical bounds.

We will continue by comparing the bootstrap sampling type 2 bounds with their theoretically motivated counterparts, the confidence bounds based on subsampling.

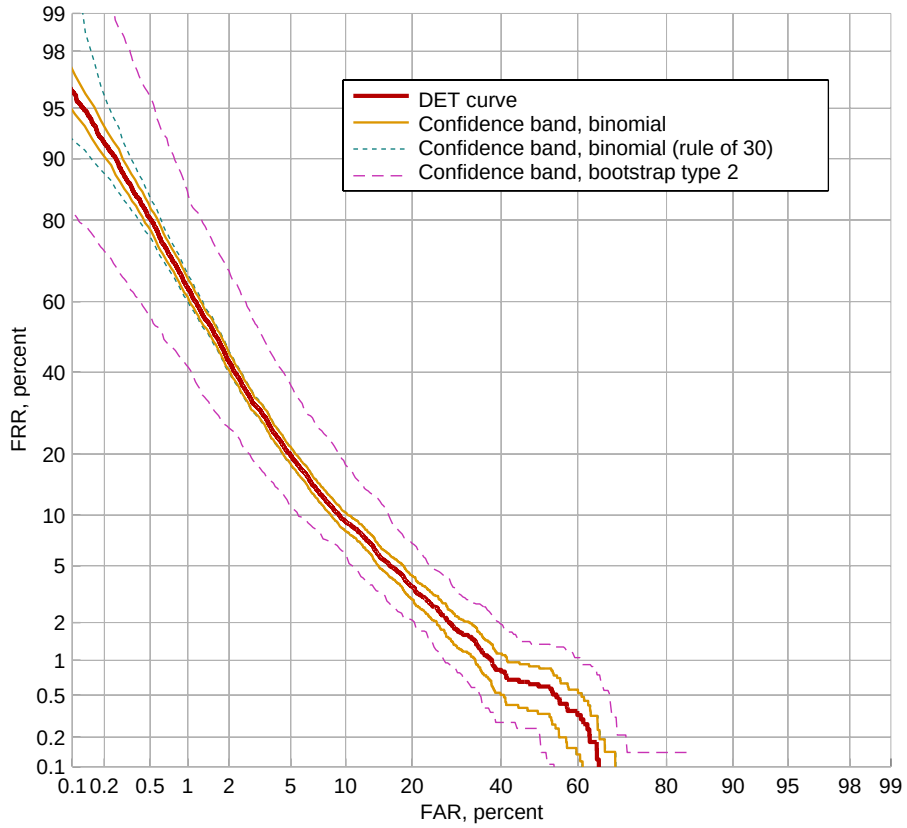

Figure 5.2: DET curve with various confidence bounds. GMM SV system (four mixtures, feature set MFCC_SD, training and test protocol "separate impostors").

5.3.2 Comparison of subsampling bounds with bootstrapping type 2 bounds

Since subsampling type 1 requires a cross-comparison test protocol, we will use the "cross-validation 30" protocol in this subsection. Both subsampling types use the same underlying statistical model for the genuine tests but differ in their model for the impostor tests. While subsampling type 1 uses Bickel's estimation for the confidence bounds of the FAR (see formula 4.32), subsampling type 2 uses the three-stage sampling model.

Figure 5.3 depicts the subsampling type 1 bounds together with the bootstrap type 2 bounds. For completeness, we also illustrate the binomial bounds.

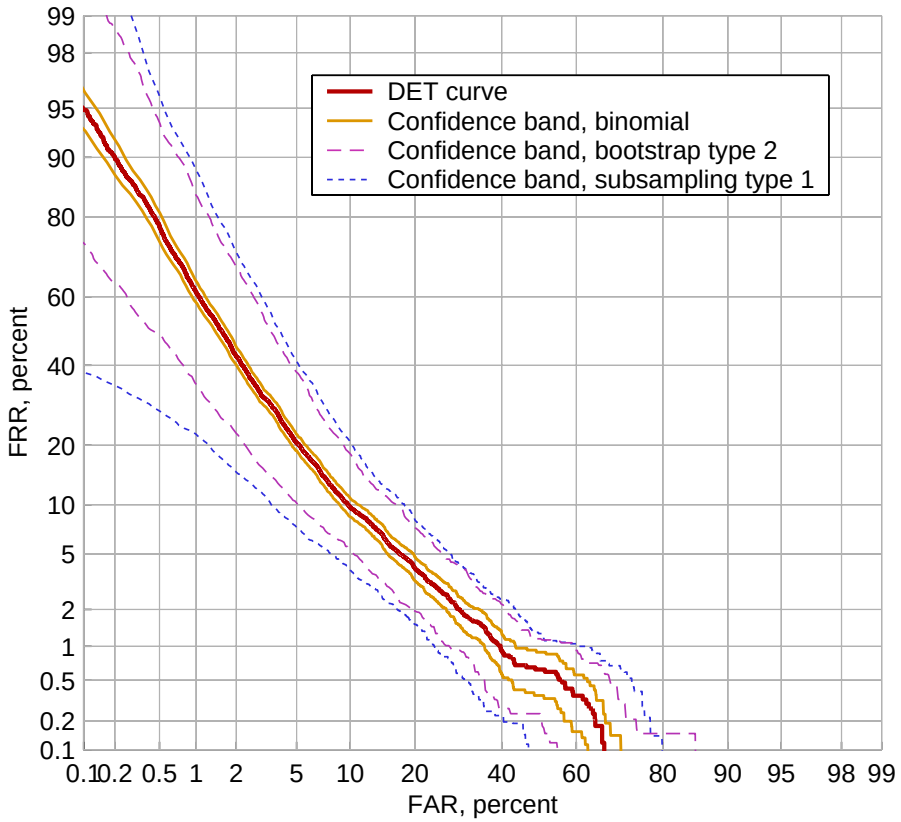

Figure 5.3: DET curve with various confidence bounds. GMM SV system (four mixtures, feature set MFCC_SD, training and test protocol "cross-validation 30").

There is a clear discrepancy between the predicted confidence band using subsampling type 1 and the empirical bands obtained via bootstrap sampling. Especially in the low FAR area, the subsampling bands overestimate the variance of the DET curve. Subsequent analysis using confidence bands with separate variation in FRR and FAR showed that the mismatch between both bands occurs mainly in the FAR.

This effect was the reason for developing the subsampling scheme type 2, that uses a modified variance estimation for the FAR. The variance of FRR is the same as for subsampling type 1 (4.29).

Further investigation showed, that the bounds predicted by the subsampling type 1 approach could be validated experimentally quite well by the approach "resampling for cross-comparison" (see section 4.4). Since the subsampling type 1 requires the cross-validation protocol, it is of limited use for general evaluation schemes.

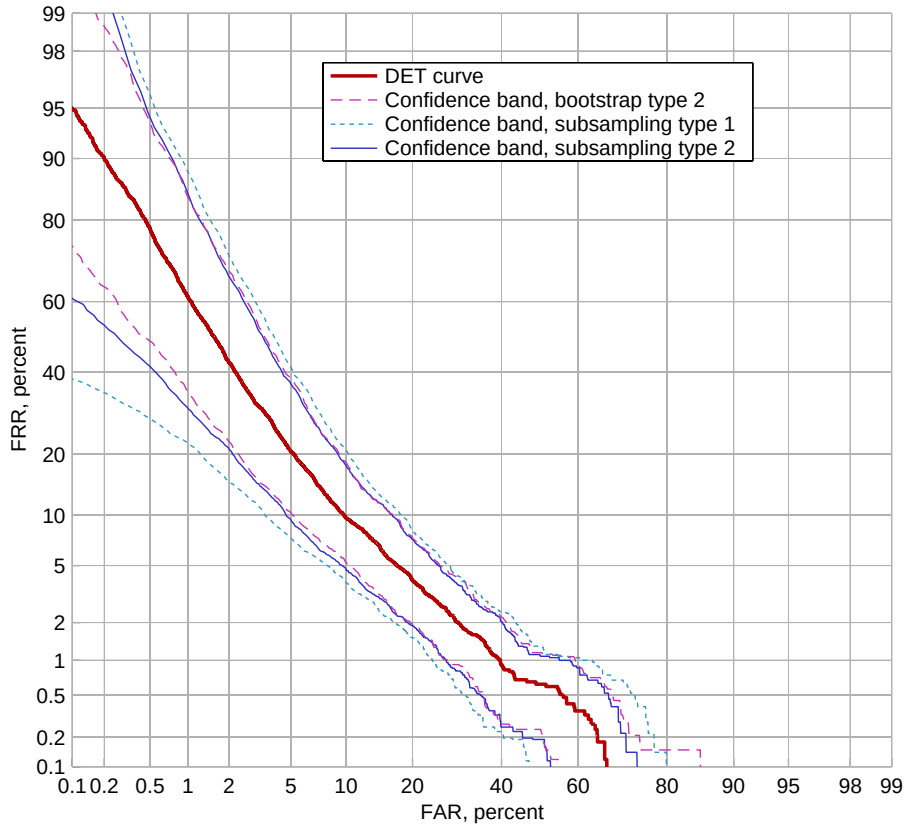

Figure 5.4: DET curve with confidence bands based on bootstrap type 2 and subsampling type 1/2. GMM SV system (four mixtures, feature set MFCC_SD, training and test protocol "cross-validation 30").

Figure 5.4 compares both subsampling types with the bootstrapping type 2. Our proposed approach subsampling type 2 predicts the bootstrap bounds much better; especially the lower empirical confidence bound of the DET curve is approximated more closely by the new type. However, some lack of fitting in the low FAR area of the DET curve remains. A great advantage of subsampling type 2 over subsampling type 1 is its applicability not only for the cross-comparison case but also for all kinds of test protocols. Especially in our evaluations, the evaluation protocol 'separate impostors' is widely used.

5.3.3 Comparison of confidence bands with McNemar test

Finally we present the application of the McNemar test (see page 28). Compared to the previous techniques of estimating the variance of performance figures it is less complex. However, as we stated already in the introduction of the McNemar test, it captures only global properties of the whole score population and does not respect the structure within the data. Thus, its properties are similar to those of the binomial approach.

Our goal here is to compare the significance between two performance figures based on the one side on the McNemar test and on the other side on the subsampling type 2.

In our example we will use two GMM SV systems with 64 mixtures and the test protocol "separate impostors". The first system is trained and evaluated using the "B-Base" scheme (called here reference system), the second one uses the "B-Q" scheme. Figure 5.5 shows the DET curve of both systems together with the confidence bounds based on subsampling type 2 of the reference system. The DET curve of the "B-Q" system lies marginally within the confidence band around

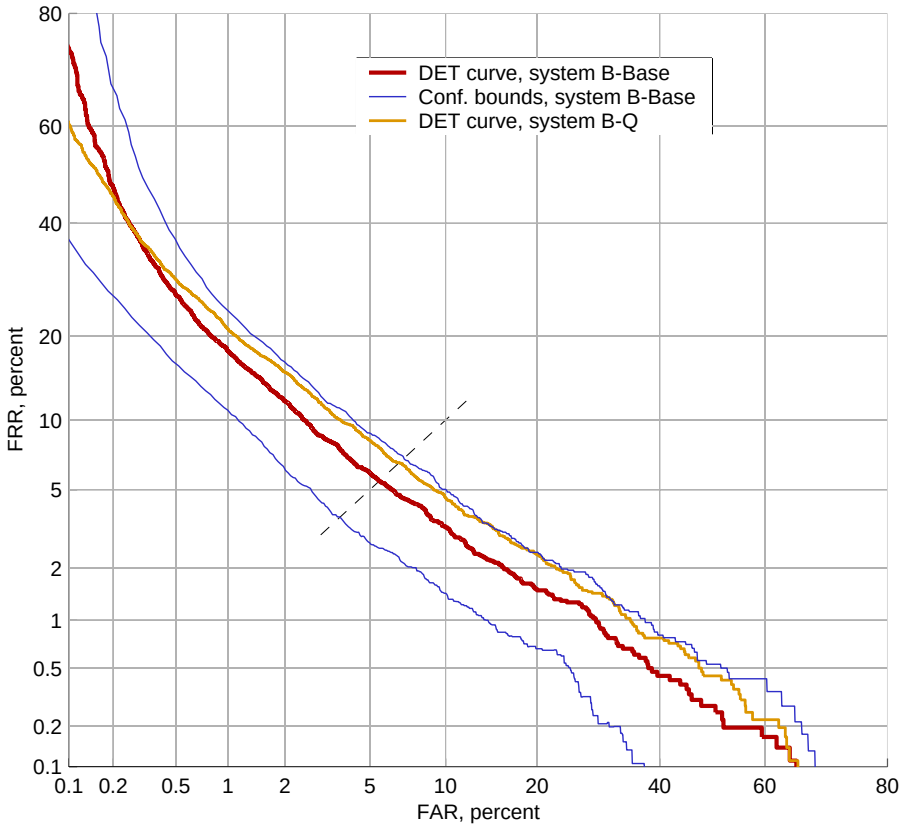

Figure 5.5: DET comparison of two GMM SV systems with 64 mixtures, feature set MFCC_SD, test protocol "separate impostors"; reference system uses the "B-Base" scheme, compared system deploys the "B-Q" scheme.

the EER point. From the graphical representation we can state, that the EER performance of the "B-Q" system does not differ from that of the reference system with a 95% confidence level. Numerical, the FRR of the second system lies within the confidence region of the FRR of the reference system, as can be seen from table 5.1. The FAR of the second system does not fall into the corresponding confidence interval. Note that the same result is achieved when comparing "B-Q" as reference system with the "B-Base" system. However, both confidence intervals cover the ERR of 6.6%.

The McNemar test for the same comparison concludes that both systems differ highly significant ($p \approx 0$; $\chi^2 \approx 594$). The result is based the total sum of differences $M_{12} + M_{21} = 3546$; hence, the required conditions described on page 28 are met. As we stated earlier in the description of the McNemar test, it does give only a very global comparison of two systems based on the number of differing decisions. The test does not take into account the situation where very similar EERs are achieved despite many different decisions of the two systems.

	B-Base	B-Q
EER in percent	5.5	6.6
conf. interv. FRR in percent	[3.1 7.8]	[4.5 8.7]
conf. interv. FAR in percent	[4.2 6.0]	[5.1 7.2]

Table 5.1: EERs with separate confidence intervals for FRR and FAR for system "B-Base" and "B-Q". Subsampling type 2 used for prediction of confidence intervals. GMM SV systems with 64 mixtures, feature set MFCC_SD, test protocol "separate impostors"

5.4 Results

As a result from the preceding experiments we can summarise:

- Confidence bands based on the binomial assumption underestimate the real variance of the DET curve since the effects of the speaker and also of the model identity are not regarded.
- Bootstrap sampling type 1 is discarded since it does not, similar to the binomial assumption, regard the structure within the score population.
- Bootstrap sampling type 2 gives empirical confidence bounds and respects the structure in the score population.
- In the cross-validation case, subsampling type 1 overestimates the bootstrap type 2 bounds while subsampling type 2 gives a much better fit of the predicted bands.
- Compared to subsampling type 1, subsampling type 2 can also be applied in the test protocol "separate impostors"
- The McNemar test underestimates the variability of the error rates as it focuses not on the performance figures themselves (like e.g. the FRR or the FAR) but on individual decisions.

Based on these facts confidence bounds for the following experiments are reported by using subsampling type 2. This type of confidence bands regards the structure within the scores and shows the best match among the investigated confidence bands with the bootstrap type 2 bounds. However it must be noted that both approaches, the subsampling type 2 and the bootstrap type 2 are based on the same underlying model with two-stage sampling for FRR and three-stage sampling of FAR. Therefore a close match between the two techniques lies inherently in their common basic assumptions.

In general the computed confidence intervals, even by the "cross-validation 120" protocol, are still too wide for comparing EER differences in the range of a few percent. Much tighter intervals would require more speakers in the database rather than a huge amount of utterances per speaker. Future data collections should respect this relationship between the size of speaker population and the achievable variance of performance figures.

Chapter 6

Robustness in speaker verification

The term "robustness" denotes a desirable property of any kind of pattern recognition system: its ability to achieve reliable recognition under different conditions compared to those of the training phase. Speaker- and also speech-recognition technology is still impeded by a performance degradation due to environmental differences between training and testing conditions. For speaker recognition systems used in telephone applications, the main cause for this mismatch is introduced by non-speaker acoustic variations due to different transmission channels, variability due to telephone handsets, different level and type of background noise. To a smaller degree, variability introduced by the speaker as described in section 4.1 causes a mismatch and thus a performance loss.

Since the VeriDat database provides extra information about the transmission channel and the prescribed level of background noise we will study robustness to these acoustic variations. We will first present some baseline experiments that reveal the effects of mismatching test conditions using a simple SV system with GMMs and a standard feature set (MFCC). We then define a measure to describe the variation of performance on different kinds of test data, the robustness coefficient, and compare it to related work from literature. Applying this new measure, we will re-evaluate the baseline experiments of mismatching test conditions. The two basic families of cepstral features, MFCC and LPCC, are compared and the effect of cepstral mean subtraction (CMS) on their robustness is studied. Finally we will present an approach of operating a SV system in always-matching test conditions by using several models trained separately on different acoustic subsets.

6.1 Baseline experiments on robustness

In the description of the training and test schemes we already mentioned the fact, that more variation in the training data leads to better verification performance. The VeriDat corpus provides two different partitions of the data, one according to the channel or network ("Fixed" and "GSM") and one according to the level of background noises ("Quiet" and "Noisy"). We will show in this section how the performance of the EER using same-gender impostors is affected by different item schemes "B-" that we presented already on page 40. We will further distinguish between *matched* and *mismatched* test conditions. In the former case the test data comes from the same acoustic subset as the training data. In the latter case different acoustic conditions are found in the training and test material. The training of the world model is always performed with the scheme "AW-Base" (see also page 40). It contains therefore all possible acoustic variations.

Mismatched test conditions

The experiments of this sections use differently trained client models evaluated with a single type of test set that contains all acoustic conditions of the VeriDat corpus. Since we regard the overall distribution of these acoustic conditions as being representative for real applications, we capture the performance degradation when client models are trained on non-representative training material. We start with a baseline experiment where the training material is also defined by a representative selection of different acoustic conditions (set "B-Base tr"). Then, the training data is selected from increasingly specific acoustic conditions: "B-Q tr", "B-N tr", "B-F tr", "B-G tr" form the first group of selectivity (using each a single dimension of the session properties) and "B-FQ tr" gives an example for even higher selectivity (using the intersection of two dimensions of the session properties).

Each of the "B-" training sets uses 20 utterances. The evaluation data "B-Base ev" is composed by all items that are not contained in any of the "B-" training sets. As we mentioned shortly, this test set can approximately be seen as containing all possible acoustic variability with the same proportions. This utterance selection scheme ensures that the evaluation is always done with the same data and that no training-on-test error can occur. Although some items in the evaluation set do always give a matched test for the specifically trained models, the complete entity of "B-Base ev" can be regarded as an evaluation under mismatched conditions. The more specific the training material is selected, the higher is the degree of mismatch between evaluation data and model.

Table 6.1 shows the result sorted by increasing EER.

Training set	EER in percent
B-Base tr	5.5
B-Q tr	6.6
B-N tr	6.6
B-F tr	15.2
B-G tr	15.6
B-FQ tr	16.2

Table 6.1: EERs of models trained with various "B-" training sets with fixed test conditions (set "B-Base ev"). SV systems uses GMMs with 64 mixtures, feature set MFCC_SD, test protocol "separate impostors", world training set "AW-Base tr".

The evaluation on the "B-Base" sets are actually matched tests and its EER is regarded as reference value here. The performance of the remaining systems using training data from the acoustical subsets deteriorates. The worst performance is shown by the system only trained with the "B-FQ tr" data. This subset is regarded as the one having the highest acoustical quality as low noise levels coincide with stable transmission lines from the fixed network.

Training on a subset based on the noise level type ("B-Q tr" or "B-N tr") shows only a slight performance degradation. In section 5.3.3 we used this data already to state that according to our selected test protocol this performance difference can not be regarded as significant.

However training on one of the two network data sets results in a high performance degradation of about 10% absolute that is statistically relevant. Obviously the acoustical mismatch seems to be greater with regard to the transmission line than to the level of background noise.

This common phenomenon is described by the term *overfitting*. It captures the fact that too specialised models can not generalize well unseen data.

Matched test conditions

The results from the previous subsection showed an increasing performance deterioration when the training material gets more and more specialized. The reverse trend can be seen when tests are done under matching conditions. Table 6.2 shows again the EER sorted by increasing values. The system trained with high-quality data only ("B-FQ tr") shows a significantly better performance than the reference system trained with "B-Base tr". The general trend is that the "Fixed" data offers better verification performance than using the "Quiet" data, that in turn is slightly better than the reference.

Training set	EER in percent
B-FQ tr	3.5
B-F tr	4.5
B-Q tr	5.3
B-Base tr	5.5
B-N tr	6.6
B-G tr	7.4

Table 6.2: EERs of models using various "B-" training sets and fixed evaluation conditions (evaluation set "B-FQ ev"). SV system using GMMs with 64 mixtures, feature set MFCC_SD, test protocol "separate impostors", world training set "AW-Base tr".

The "B-N" and the "B-G" systems show a worse performance than the reference system. The test data of both acoustical conditions might contain more variability compared to the "B-F" and the "B-Q" case. Using the same amount of training material this variation can not be captured very well and thus the performance decreases compared to the reference system.

A common rule of pattern recognition shows up here: the more variable the data is, the lower the recognition performance gets when the complexity of the classifier and the amount of training data is kept constant.

6.2 Robustness coefficient

In this section we will state the need for a simple term that can be used to compare the performance behaviour of a SV system in matching and non-matching test conditions. We introduce an additional performance figure, the *robustness coefficient*, and compare our definition with related work. Finally we exemplify its application on our baseline SV system using the standard feature set MFCC_SD.

6.2.1 Motivation

Our example performances in the mismatched case from the last section showed high variability depending on the training set. When using a test set with a representative distribution of the four different acoustical conditions (e.g. "B-Base ev"), the effects of the different performances on the disjoint data subsets are merged into a single figure. An alternative could be reporting the performance on two subsets generated from either the two acoustical partitionings "Fixed" vs. "GSM" and "Quiet" vs. "Noisy", or on all four combinations of them. However, this results in more figures that have to be compared.

6.2.2 Definition of the absolute and relative robustness coefficient

It seems therefore desirable to define a term that describes the spread of the EER on the different subsets. We propose an *absolute robustness coefficient* denoted as *RC*. It is defined in formula 6.2 as the reference performance of the system on representative test data (denoted as EER_r) minus the mean deviation between the reference performance and the performances on acoustical motivated subsets of the test data (generically denoted as EER_{A_i}). The division into disjoint subsets can be done in any domain that can affect the performance of the SV system.

We will later use two variations of the robustness coefficient: an absolute version as described above and a relative one. The latter is received by normalising with the reference performance EER_r (see formula 6.4). The relative robustness coefficient is denoted using lower case letters as *rc*.

The absolute difference between the performance on a particular subset A_i $(i = 1 \dots N)$ of the total set $\mathcal{A}$ and the reference performance is generically denoted as

$$\Delta(EER_{A_i}) = \left| EER_{A_i} - EER_r \right| . \tag{6.1}$$

The absolute robustness coefficient corresponding to the set $\mathcal{A}$ is defined by

$$RC_{\mathcal{A}} \quad = \quad EER_r - \frac{1}{N} \sum_{i=1}^{N} \Delta(EER_{A_i}) \tag{6.2}$$

In this section the reference performance is always measured using the "B-Base ev" set. The performance EER_{A_i} on the acoustical subsets are denoted by replacing the symbol A_i by the matching identifier of one of the subsets, namely *F*, *G*, *Q*, *N*, *FQ*, *FN*, *GQ* or *GN*.

In case of dividing the data into "Fixed" and "GSM" recordings, the corresponding absolute robustness coefficient is defined by

$$RC_{FG} \quad = \quad EER_r - \frac{\Delta(EER_F) + \Delta(EER_G)}{2} \tag{6.3}$$

where EER_F and EER_G denote the EER on the "Fixed" respectively "GSM" subset of the test data "B-Base ev".

The corresponding relative robustness coefficient is defined by dividing by the reference EER:

$$rc_{\mathcal{A}} \quad = \quad \frac{RC_{\mathcal{A}}}{EER_r} \tag{6.4}$$

Please note that a system showing no performance variation when testing with different mismatched test sets is defined to have a relative robustness coefficient of 1.0. Typically real world systems have a robustness coefficient below 1.0. Lower values indicate systems that are sensitive to variations in environment and noise level. Even negative values can occur when the mean deviation from the reference performance is greater than the reference performance itself. This may happen especially for systems having a very small reference EER.

6.2.3 Extension to weighted performance on subsets

When computing the mean deviation between the reference performance and the performances on the acoustical subsets in formula 6.2, we inherently assumed that each contribution to the

sum is weighted equally. For some application it might be sensible to weight the performance difference for some acoustical conditions more than others. E.g. a infield test might reveal that certain conditions are more frequent than other. A more refined absolute robustness coefficient can be described using weighting factors $w_1, \ldots, w_N$ which are typically between 0 and 1:

$$RC_{\mathcal{A}}(w_1, \ldots, w_N) \quad = \quad EER_r - \frac{1}{N} \sum_{i=1}^{N} w_i \Delta(EER_{A_i}) \; . \tag{6.5}$$

The relative robustness coefficient is computed similarly to formula 6.4.

6.2.4 Related work

To our knowledge there is only one other approach for defining a figure capturing the robustness of a speaker verification system. Zilca [54] introduces two robustness figures for comparing SV systems trained on data from two different types of telephone handsets. He uses the NIST-2000 speaker recognition evaluation data [55] that includes a labeling of the handset type used (electret vs. carbon button microphone) denoted as 'et' and 'cb'. This partitioning of the data results in four combinations of training and test conditions: 'et/et', 'cb/cb', 'et/cb' and 'cb/et'. While the former two combinations represent matching conditions, the latter ones represent an increasing mismatch. Especially the system 'cb/et' shows the worst performance due to the low quality of the carbon button data.

The robustness figures capture the relative mean degradation of the EER between matching and mismatching conditions (R) and the relative EER degradation between the two possible matching conditions (Q). The first robustness measure is defined as

$$R = \frac{R_1 + R_2}{2} = \frac{1}{2} \left(\frac{EER(cb/et)}{EER(cb/cb)} + \frac{EER(et/cb)}{EER(et/et)} \right) . \tag{6.6}$$

It is the average of the relative performances on matching and mismatching conditions, R_1 and R_2.

The second robustness measure, Q represents the system sensitivity to the type of handset in matching conditions:

$$Q = \frac{EER(cb/cb)}{EER(et/et)}. \tag{6.7}$$

Both measures are used by Zilca to compare SV systems with regard to sensitivity to the handset type and to handset type mismatch.

Compared to our approach, Zilca's performance figures allow a detailed separation of the effects introduced by variability in the data. In turn, they require training of different systems and more performance figures get involved into a comparison.

Our definition of the robustness coefficient is based on the assumption, that a fixed distribution of conditions occur in practical applications and the performance on this data set can be regarded as reference. Performance deviations from this reference due to various partitionings are captured in a single performance figure. The concept is not limited to dichotomous data but can easily be extended to a larger number of data splits.

6.2.5 Application to mismatched test conditions

We will revisit the performance figures in the mismatched case from table 6.1 and compute all three types of robustness coefficients. The data is provided in table 6.3.

Training set	EER in percent	rc_{FG}	rc_{QN}	rc_{FGQN}
B-Base tr	5.5	0.84	0.83	0.78
B-Q tr	6.6	0.85	0.77	0.77
B-N tr	6.6	0.99	1.00	0.98
B-F tr	15.2	0.50	0.89	0.49
B-G tr	15.6	0.63	0.94	0.64
B-FQ tr	16.2	0.50	0.93	0.47

Table 6.3: EERs and relative robustness coefficients of models using various "B-" training sets with mismatched test conditions ("B-Base ev"). SV system uses GMMs with 64 mixtures, feature set MFCC_SD, test protocol "separate impostors", world training set "AW-Base tr".

Generally, the robustness coefficient describes the performance spread of the different systems quite well. E.g. the "B-F" and the "B-G" systems have a lower rc_{FG} compared to the "B-Q" and the "B-N" systems. The same behaviour can be depicted from the EERs, where the systems specialised to a network type suffer more under mismatched conditions. A comparison of only the robustness coefficients can be done more easily.

The higher variation when dividing according to the network type instead of the noise level that we found in the last section, displays again that the rc_{FG} figures reach lower values than the rc_{QN} values. The rc_{FGQN} figures approximate the smaller value of the former two robustness coefficients. The sum over the relative performance deviations in formula 6.2 captures also variations within one domain in the robustness coefficient. We will exploit this convenient property and use this coefficient most of the time when measuring robustness coefficients for the comparison of SV systems.

Table 6.3 also unveils a surprising behaviour of the "B-N" system. Although its performance is slightly worse compared to the reference system, it exhibits virtually no performance variability due to mismatched test conditions and thus turns out as highly robust.

According to these results, using noisy training data renders a SV system more robust to various acoustical conditions. The overall performance of the "B-Q" system is the same but its rc figures of all three types are about 0.2 units lower. The variability in the noisy data seems to widen the Gaussian mixtures in the GMM in such a way, that also the variation induced by different networks can be handled without performance loss.

6.2.6 Further application to feature comparison

The application of the robustness coefficient is not restricted to the comparison of systems using different training sets. We will use this figure to characterise different feature sets. It can be seen as an additional criterion when selecting training strategies and key system parameters. Some applications might allow a lower performance but require a more stable and thus more reliable performance under unpredictable acoustic conditions.

6.3 Robustness of cepstral features: MFCC and LPCC

We will apply the robustness coefficient for an examination of the two mainly used families of cepstral coefficients, MFCC and LPCC. For a further description of both types of cepstral

coefficients we refer to chapter 3.2. A common technique for increasing the robustness to channel variations of these coefficients is also included in the investigations: the cepstral mean subtraction (CMS). In speech recognition systems this normalisation improves performance significantly, while in speaker recognition systems, beside a performance gain in mismatching test conditions, a performance drop is experienced with CMS when training and testing is done under matching conditions.

We will examine the robustness of both cepstral coefficient families using our standard evaluation set "B-Base ev". The robustness can thus be measured on both partitions, the channel domain and the noise level domain.

The SV systems used here are again based on GMMs, this time using a varying number of Gaussian mixtures starting with a single mixture and going up to 512 mixtures by powers of two. This allows us to study the effect of the robustness under different degrees of complexity of the classifier. Both world and client models have the same number of mixtures.

Finally we will cite some work from literature dealing with performance and robustness differences between LPCCs and MFCCs.

LPCC with and without CMS applied

Figure 6.1 depicts several performance parameters under varying number of Gaussian mixtures: the EER measured from the DET curve using the "separate impostors" test protocol and same gender impostors, and its corresponding relative robustness coefficients. For easier illustration of the number of mixtures on the abscissa, a logarithmic scale using only the exponents of the powers of two is applied.

Increasing complexity of the classifiers allows an increasing performance as expected (figure 6.1 a). Applying CMS normalisation shows only an improvement for two, four and 256 or more mixtures. In general, LPCC features seem to be more robust to different noise levels than to different channels (figure 6.1 b). The estimation of the poles can still be performed reliably from speech with additive noise. However, differences in the channel characteristics seem to influence the position of poles and changes therefore the cepstral coefficients.

This general trend is still true when applying CMS. The robustness coefficient on the network type, rc_{FG}, is approximately increased by 0.15 units with CMS normalisation for 4 or more Gaussian mixtures. Under CMS usage rc_{QN} deteriorates slightly which decreases the difference between the robustness with regard to the network type and with regard to the background noise drastically. CMS for LPCC seems therefore efficient in reducing variability due to different channels while degrading the robustness to the noise level only slightly.

The largest gain for rc_{FG} due to CMS can be found in a midrange complex modelling using 16 to 128 Gaussian mixtures.

MFCC with and without CMS applied

The performance of a SV system using MFCC shows a quite similar picture in comparison with employing LPCC features (figure 6.2 a). Without CMS, the EER performance reaches approximately the same level when using 512 mixtures. However, the improvement builds up more slowly with increasing complexity compared to the LPCC case. CMS gives a consistent performance gain when deploying 8 or more Gaussian mixtures.

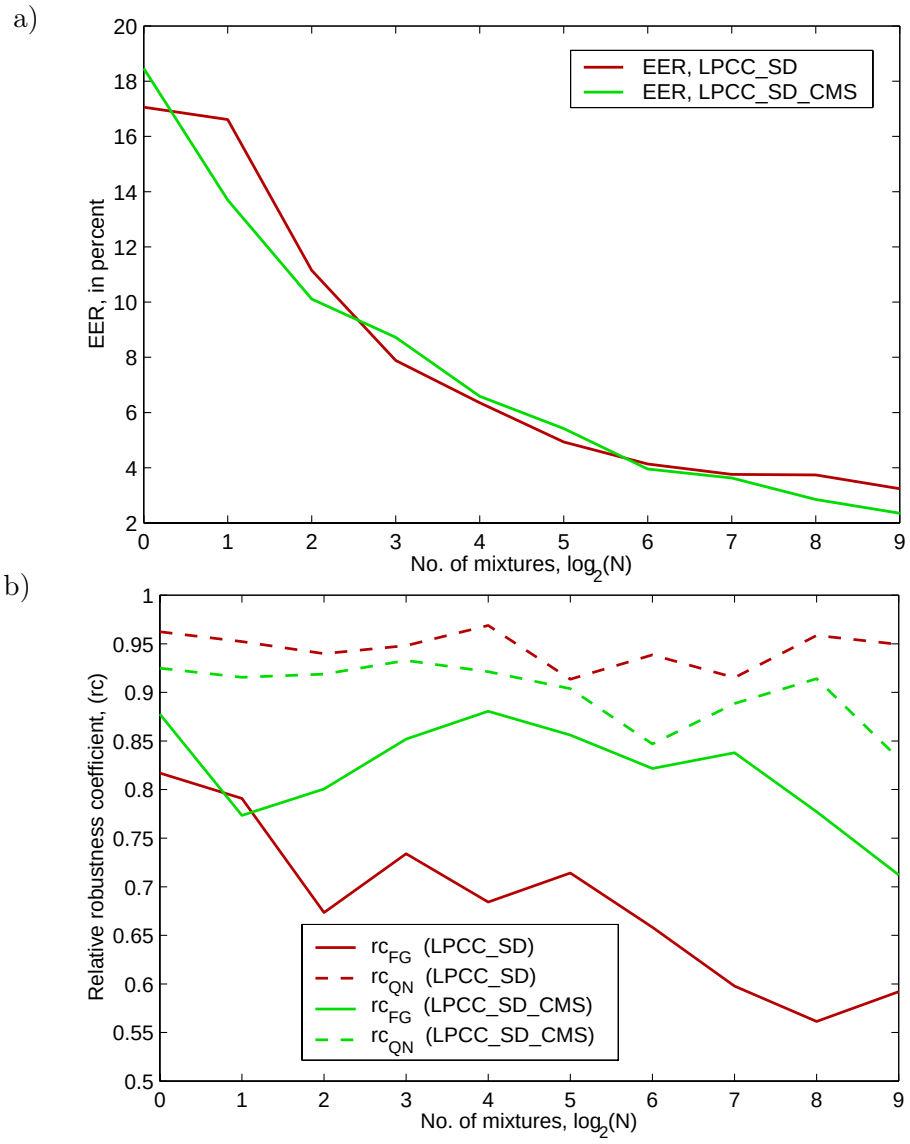

Figure 6.1: Performance of GMM SV system with varying number of Gaussian mixtures using the LPCC_SD and LPCC_SD_CMS feature sets. EER measured from DET curve using the "separate impostors" test protocol and same gender impostors on top (a) and relative robustness coefficients on the bottom (b).

The robustness coefficients show a reversed situation compared to the LPCC case (figure 6.2 b): MFCCs are more robust to channel variation than to variation of noise level. Though this behaviour can only be observed with a reasonable amount of mixtures and depicts not as clearly as the respective behaviour of the LPCC features. Commonly for both feature sets, CMS causes the same changes in the relative robustness coefficients: the rc_{FG} increases by about 0.1 to 0.15 units while the rc_{QN} shows only a slight deterioration for 8 or more Gaussian mixtures.

To conclude we state that CMS normalisation mainly reduces performance variability on different channels, as it is intended to. It can not tackle variation due to additive noise. Compared to MFCC features, CMS with LPCC still results in variability on the channel as indicated by higher robustness coefficients. This is probably the reason why additional robustness techniques e.g. feature warping are applied mainly for LPCC features.

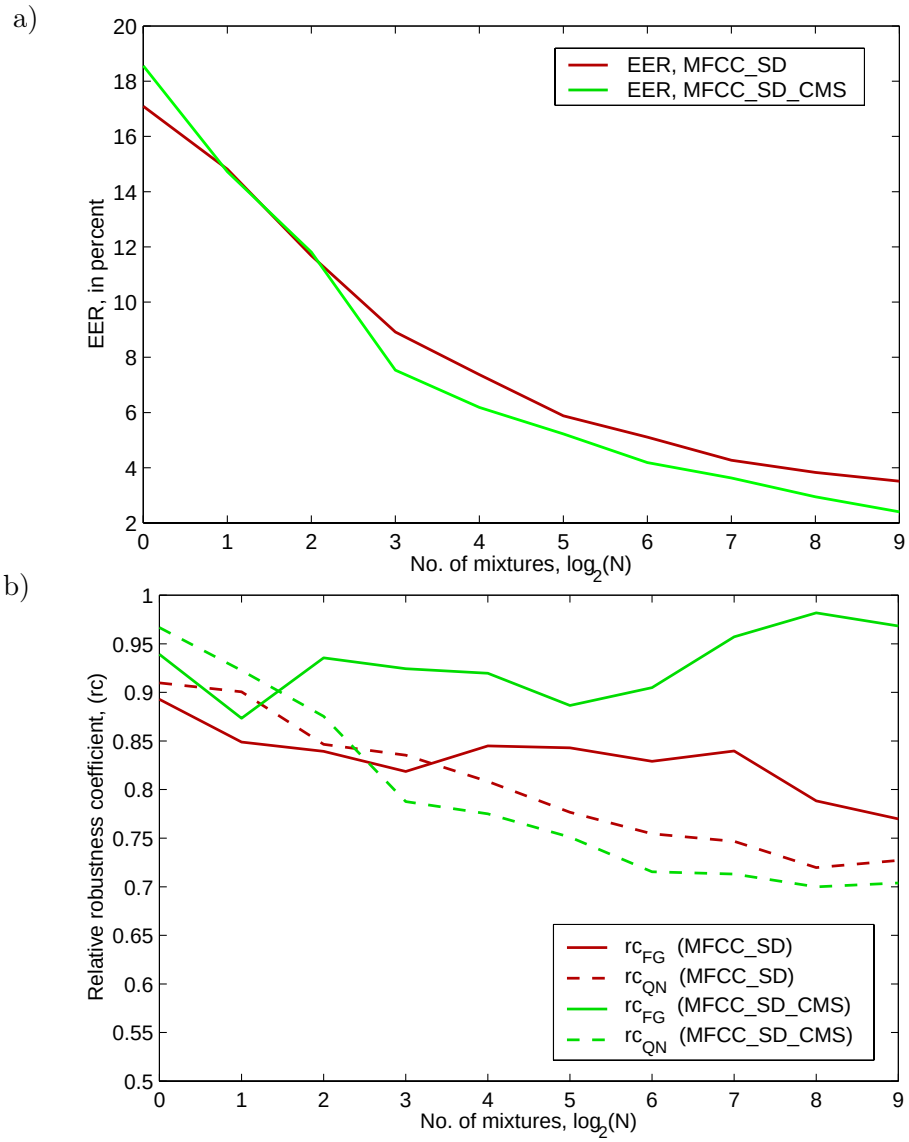

Figure 6.2: Performance of GMM SV system with varying number of Gaussian mixtures using the MFCC_SD and MFCC_SD_CMS feature sets. EER measured from DET curve using the "separate impostors" test protocol and same gender impostors on top (a) and relative robustness coefficients (b).

Related comparisons of LPCCs with MFCCs

Some reports deal with the robustness and performance differences between LPC-based cepstra (LPCCs) and FFT-based (MFCCs) cepstra. The main conceptual differences have already been presented in section 3.2. First, the Mel-based frequency transformation is usually only applied to MFCCs. The second difference is, that LPC-based cepstra are derived from an already estimated spectral shape given by the linear predictive coefficients.

In [56], it is shown by Kuitert and Boves, that cepstra derived from spectra processed with Mel-based filters perform slightly better in a speaker recognition task. This applies also to LPC-based cepstra. The authors argue that probably the smoothing effect in the high frequency area reduces random variation in that frequency area.

For the CAVE project, a speaker recognition research program using telephone speech data, Bimbot et al. [57] reports a performance advantage of LPCCs over MFCCs for simple HMM topologies for their sub-word models. Complexity of the models is measured by the product of

the number of HMM states per phoneme and the number of Gaussian mixtures per HMM state. However, the performance gain disappears for higher complexity models. No comparison is taken out between the two cepstra families for environmental effects.

6.4 Robustness by deploying separate acoustic models

The performance figures based on the matching test conditions from table 6.2 were the motivation for the experiments described in [53].

These investigations were setup in order to show if information about the used telephone network or the level of background noise could be successfully exploited by using two distinctive world and/or client models. The option is to use either one model trained with the complete data set or to train two specialised models with separate data. The division of the data could be done e.g. in the domain of the telephone network which results in two separate models, one for the "Fixed" and one for the "GSM" data. Assuming that the type of telephone network is known to the speaker verification system, it can select the appropriate model when operating in the verification mode (so-called *cheating experiment*). The models are thus always tested in a matched condition.

The SV system used in [53] consists of a sub-word HMM system using single Gaussian mixtures and the feature set LPCC_SD. The results presented in this section are based on a GMM system with 64 Gaussian mixtures and the feature set MFCC_SD. As we already discussed in the previous section, the MFCC coefficients seem to be more robust to mismatch in channel distortions than to additive noise while the situation is reversed for the LPCC coefficients. Thus we would expect different outcomes of the former and current experiments when performing cheating on either the network or on the noise level.

The cheating experiments were ensured to be fair by using equal amounts of training data for all three types of models (complete model, first and second model from split). Table 6.4 describes the training sets applied in the cheating experiments. The "C-Base" experiment is very similar to the "B-Base" experiment from page 56, however the world model is trained using the "BW-Base tr" training set. In case of splitting, denoted by the split type (e.g. F/G), the models are trained separately using two training sets. A two-digits postfix describes the number of parts used for the world model and the client model. E.g. "C-Q/N 1/2" denotes the cheating experiment using splitting along the "Quiet" vs. "Noisy" domain, where only the client models are trained separately while the world model is trained using the joint data set.

All genuine tests are done using the "B-Base ev" set while impostor tests are again performed with the full set of utterances.

The results from the cheating experiments are displayed in table 6.5 for splitting along the network type (F/G) and 6.6 for splitting along the noise level (Q/N). The number of models indicate whether the joint data set (one model) or a split data set (two models) is used for training.

Although sub-word HMMs with a different feature set were used in [53], the main effects are the same compared to the current results. Splitting of any type in the network domain causes a performance degradation. We would have expected an advantage for the system using the split for both the world and the client model since it deploys the most sophisticated setup using matching test conditions for both models..

While splitting in the noise level domain, the results tend to match our expectations. A slight performance gain is achieved when splitting the world model. In the remaining case, splitting

Exp.	**World Model(s)**	**Client Model(s)**
C-Base	single model: • BW-Base tr (all conditions) 1260 recordings	single model: • B-Base tr (all conditions) 20 recordings
C-F/G 2/2	two models: • Fixed, BW-F tr, 1260 recordings • GSM, BW-G tr, 1260 recordings	two models: • Fixed, B-F tr, 20 recordings • GSM, B-G tr, 20 recordings
C-F/G 1/2	single model	two models
C-F/G 2/1	two models	single model
C-Q/N 2/2	two models: • Quiet, BW-Q tr, 1260 recordings • Noisy, BW-N tr, 1260 recordings	two models: • Quiet, B-Q tr, 20 recordings • Noisy, B-N tr, 20 recordings
C-Q/N 1/2	single model	two models
C-Q/N 2/1	two models	single model

Table 6.4: Training parameters for the baseline and the cheating experiments (key word explanations see text). GMM SV systems with 64 mixtures, feature set MFCC_SD, test protocol "separate impostors".

only the client models, the performance degrades. Compared to the former experiments using LPCC features [53], this split type was the only one showing a performance gain. Surprisingly, this behaviour is not completely consistent with our expectation.

A split in the network domain will result in a performance drop compared to a noise level split. Since MFCCs show a higher robustness for channel variations compared to variations with regard to the noise level (see plots with 64 mixtures in figure 6.2) we would not have expected the behaviour given by the performance figures from table 6.5 and 6.6. The smaller relative robustness with regard to the noise level could lead to the assumption that cheating on the noise level will give a larger performance gain compared to cheating on the network domain. In addition we would have expected performance improvements in both cases.

No. of world models	No. of client models	
	1	2
1	5.5	6.7
2	5.6	5.9

Table 6.5: Results for the baseline and the cheating experiments with splitting along the network type ("Fixed" vs. "GSM").

No. of world models	No. of client models	
	1	2
1	5.5	5.8
2	5.4	5.4

Table 6.6: Results for the baseline and the cheating experiments with splitting along the noise level ("Quiet" vs. "Noisy").

Please note that all observed performance differences are far below the statistically relevant gains that can be measured with the applied test set.

A possible cause for the inconsistent success of using separate acoustic models might be ascribed to the performance in the matched case for different acoustic subsets (table 6.2). In both partitions one of the specialised systems performs weaker than the reference system while the other is performing better than the reference. The low performing part of the model pair might conceal the advantage of the splitting technique and thus influences the overall performance.

As a final result we state that training separate acoustical models does not give a statistically relevant performance gain. Especially in the network domain where we expected an improvement due to the high sensitivity of the MFCC features for a mismatch, the performance deteriorates with any modification of the base system. Although, due the low number of training and testing data, the differences can not be counted as statistically relevant; the findings indicate that training a single model with data taken from all acoustic conditions gives the best achievable performance.

Chapter 7

Impact of GSM coding on speech cepstrum

In this chapter, we will investigate the effects of the GSM coding on speech represented by LP-based cepstral coefficients. First, we will present the main properties of the GSM speech coding system and its different extensions. Based on this information we state a hypothesis about the type of GSM speech codec that might have been used in the VeriDat database.

We describe two simple simulation models that we developed in order to artificially reconstruct some effects of the GSM transmission of speech. The first model includes only the distortion of the speech coder/decoder combination in the transmission chain while the second model also takes into account the effects of errors during the radio transmission. Fixed line recordings are processed by these different simulation models and the results are stored in so-called *transcoded recording sets*.

Next, the two main speech degradation effects, linear channel and additive noise, are presented and their impact on LP-based cepstral coefficients is discussed. For the investigations in this chapter here, a subset of the VeriDat data is created: fixed line recordings, time-aligned GSM transcoded recordings and a manually created phonetic segmentation are joined in a stereo corpus. Using the stereo data, we explore the effects of the GSM speech coding in the cepstral domain by grouping segments based on their associated phonetic class. We present the transformation of the cepstral clusters both qualitatively by using scatter plots and numerically using the Bhattacharyya distance.

The results from this chapter will be used in the adaptation scheme for speaker verification models that we will present later in chapters 8 and 9.

7.1 GSM speech coding

The Global System for Mobile communications (GSM) is the standard digital mobile system used throughout Europe and many other countries. During the time since its commercial introduction in 1991, several speech codecs for GSM have been standardized by the European Telecommunication Standards Institute (ETSI). As the data rate of the output stream used in fixed line telephony (64 kbps) is too high, the signal must be transcoded to a lower data rate feasible for transmission over a radiolink. The channel bit rate in the GSM standard is 22.8 kbps. Several lossy speech codecs for this target rate have been evaluated by the GSM group on the basis of subjective speech quality and complexity.

Figure 7.1 depicts the different stages involved in the transmission of speech with GSM. The speech coder and decoder that we present in this section more detailed are located at the start and the end of the complete processing chain. We will later come back to this figure when describing the channel coding and the interleaver properties.

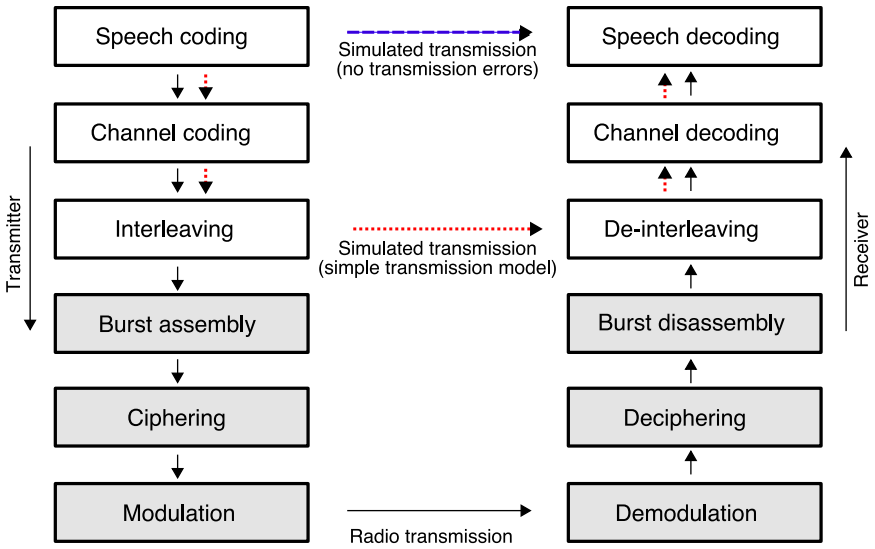

Figure 7.1: Processing chain for transmission of speech data with GSM. Only the stages displayed with white boxes are relevant for the further explanations of the GSM transmission simulations. Two simplified simulations of the complete processing are used here: a simulation of the speech coder/decoder effects only (using the path plotted with a dashed line) and a simplified simulation of the transmission with effects due to bit errors (using the path with a dotted line).

The main GSM speech codec that has been available since the very beginning of the GSM standard is the Full Rate codec (FR), described in the ETSI document [1] GSM 06.10 [58]. It is also named after its main components: RPE-LPT (Regular Pulse Excited - Long-Term Prediction). It is based on a vocoder using LP-analysis of the signal with a sophisticated encoding of the residual signal. It operates at a data rate of 13 kbps. A reference implementation in C [59] is available and also various speech streaming utilities apply it for signal compression e.g. in OpenH323 [60].

The currently used speech codec is called Enhanced Full Rate (EFR) and has been introduced in mobile devices in the end of the year 1997. It was developed by Nokia in cooperation with the University of Sherbrooke (Canada) and first standardized by the ETSI in 1995. Its description is provided in GSM 06.60, the reference implementation can be found in GSM 06.53. Nowadays it is the default speech codec in current GSM networks and mobile devices.

The VeriDat database contains no information about the speech codec used in the GSM recordings. The first GSM EFR capable mobile devices became available in 1998, the first networks have been updated in mid-1998. We assume that it took some time for this new technology to reach a noticeable market share. Thus most of the recordings might still be done using the old GSM FR codec as they took mainly place in autumn and winter 1999/2000. However, this property depends on the mobile device the speaker used and the area where the recording took place. These influences might even change from session to session, if a speaker selected among different cellular phones.

Due to these considerations we take into account in most of our following experiments only the GSM FR codec.

[1] All GSM documents cited here are publicly available from http://www.etsi.org

7.1.1 Common properties of GSM speech codecs

All three codecs support a technique called Discontinuous Transmission (DTX). Its function is to suspend radio transmission during silence periods. If we take into consideration that a person normally speaks less than about 40 to 50 percent of the total time during a conversation, the transmitted data can be reduced by approximately the same amount. DTX increases thus the capacity and reduces power consumption of the mobile device.

It relies on two main components: a voice activity detection (VAD) that determines the speech and non-speech parts of the signal and a comfort noise generator. The latter creates a minimum background noise called comfort noise in the target device during speech pauses in order to give the user the impression that the connection is still alive. Otherwise, total silence parts in between the speech segments will mislead the dialog partner to the impression that the connection is interrupted.

7.1.2 Simulation of the transmission channel

In order to take into account also the effects of the transmission of the coded speech over the GSM network, we provide for some experiments a simulation of the GSM radio channel. Typically for such simulations of binary data transmissions is an *error insertion device* (EID) which simulates transmission errors. A simple but commonly used model for a transmission channel is the *Gilbert Elliot channel* (GEC) model (see [61, chapter 4]). It assumes that the channel has two states, labeled 'G' for 'good' and 'B' for 'bad' where each state is associated with a probability for a bit error during transmission. Normally, the good state of the channel (P_G) is characterised by a low bit error rate (typically $P_G = 0$) while the bad state of the channel shows a high bit error rate, typically $P_B = 0.5$. The transition probability from the state 'bad' to the state 'good' is given by Q, while the transition in the opposite direction is denoted by P. This model allows to include a correlation between bit errors because it does not use a single and constant bit error rate that applies for all bits in the transmission.

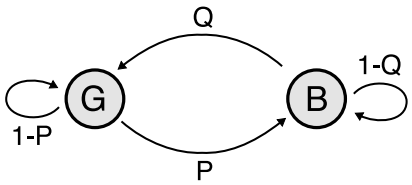

Figure 7.2: Gilbert Elliot channel model. Two states represent the two different conditions of the channel: a good condition ('G') and bad condition ('B'). The transition probability to the opposite state is given by P respectively Q. Each state is associated with a bit error rate P_G and P_B.

The mean bit error probability (BER) is given by

$$\text{BER} = \frac{P}{1-\gamma} P_B \; + \; \frac{Q}{1-\gamma} P_G \tag{7.1}$$

with

$$\gamma = 1 - (P + Q). \tag{7.2}$$

The term γ denotes the correlation between bit errors; $\gamma = 0$ indicates a nearly random error channel, while $\gamma = 1$ implies a totally bursty channel.

To simplify the model for the following calculations, we assume a totally error-free channel in the good state ($P_G = 0$) and a random channel in the bad state ($P_B = 0.5$). Thus the two remaining

degrees of freedom in this model are the mean bit error probability (BER) and the correlation term (γ).

Our implementation of the channel model uses the EID routines of STL-ITU utilities ([61, chapter 4]). The optional frame erasure probability also provided by the EID routines is not used here. The channel bits are taken from the interleaver and processed frame-wise (eight times 57 bits = 456 bits).

Here we assumed that the 57 bit sub-blocks are concatenated seamlessly and all eight bursts in the frame are occupied with data from the same transmitter. This simplification does not take into account bursty errors affecting the data beyond a single sub-block. We assume that this effect does not play a major role.

7.2 Simulation models for GSM recordings

We stated already earlier that our adaptation scheme uses simulated GSM recordings that are generated from genuine fixed line recordings. Two different simulation models for GSM recordings and their corresponding data set of simulated GSM recordings are presented in this section. In the diagram of the GSM processing stages (figure 7.1) we plotted two different paths that could be regarded as simplifications of the complete path through all stages.

7.2.1 GSM simulation without transmission effects

In a first approach using a simple simulation, we will simulate a GSM recording by transcoding fixed line recordings with a GSM speech codec and decode the speech data directly after encoding again. This processing chain is depicted in figure 7.1 with the dashed path. The transmission between transmitter and receiver is assumed to be error-free i.e. a perfect radio link is available.

For this model we have the choice between the FR and the EFR speech codec. We will denote these simulation data sets by *sim G FR* and *sim G EFR* respectively. In most of our experiments, we will employ the *sim G FR* set in our experiments.

7.2.2 GSM simulation including transmission effects

A second, more advanced simulation takes also the effects of the transmission channel into account. The corresponding path through the transmission stages from figure 7.1 is depicted by the path marked with dotted arrows.

In order to keep the simulation computationally simple we make the assumption that the transmission channel shows only a moderate bit error rate. We assume further, that then only the class II bits suffer from bit errors while the error protected parts in the data frame (class Ia and class Ib bits) can still be recovered completely. Thus we avoid the complex simulation of encoding and recovering the class Ia and class Ib bits and only apply the error model of the transmission channel to the class II part of the data block. The remaining parts are bypassed unaffected from the transmission channel because we assumed that distorted bits in these parts can be corrected.

Including the channel coding stage and the interleaver stage into the model is necessary when the bit errors are simulated by the GEC model as in our case. The parameter γ of the GEC model describes the correlation between bit errors (burst errors). Real radio link channels show a

significant amount of bursty errors that should be regarded in the simulation. Both the channel coder and the interleaver reorder the bits of the speech coder parameters. By spreading the bits of a single speech data frame, the effect of error bursts are reduced and the corruption of a complete speech frame is less probable.

In a GSM simulation without these two stages the effect of burst transmission errors would be estimated too high since they could be concentrated only on few parameters of a speech frame.

The remaining processing stages of figure 7.1 are assumed to be transparent in terms of the effects of transmission errors. However this is not proven, especially for the ciphering/deciphering stage.

The two parameters of the transmission channel model are set according to a bit error statistic for typically occurring quality levels (see [58, table A1.2]). A bit error rate (BER) of the unprotected class II bits of 8.3% gives an error rate of 0.61% for the class I bits. Our assumption that the class I bits can be fully recovered is nearly fulfilled.

A lower quality channel with a class II BER of 13.0% results in a class I BER of 4.1%. Here, the assumption is violated. However, such high BERs (i. e. low channel qualities) are not typically for a normal coverage of the GSM network (see [58, section A.1.2.1]).

We will later show experiments with transcoded recordings using these two different transmission channel simulations. As speech codec we selected only the FR codec and assume that the same results will be achieved with the EFR codec. A high quality transmission with $BER = 8\%$ is used for the recording set denoted as *sim G FR C-0.08* while a low quality transmission is used for the transmission set *sim G FR C-0.13*. The letter C with its parameter denotes the simulated transmission channel with its bit error rate.

7.3 Transcoded recording sets

The different GSM simulation models have been used to create additional recording sets from the fixed line recordings of the VeriDat corpus. Thus the simulated GSM data sets comprise the same material as the fixed line recordings. In addition, only minimal time shifts due to different handling of incomplete frames at the end of the recordings occur. This fact allows us to apply the same temporal information about the recordings e.g. segmentation information both to the genuine fixed line recordings and to the simulated GSM recordings.

As we already mentioned briefly in the introduction we try to simulate the situation when a recording takes place using a GSM cellular phone. Our first assumption here is that GSM recordings differ compared to fixed line recordings only in the presence of additional speech codec artifacts and optionally in additional effects of bit errors in the radio transmission. Two transcoded recording sets have been generated by the shortcut link between speech coder and decoder (no transmission errors) using the reference implementations of both the FR and the EFR GS codec. The more realistic transcoded data deploying the additional stages for error protection (with transmission errors) was created only for the FR speech codec.

We used the provided reference implementations for the FR and the EFR codec to simulate the speech coding/decoding stages from figure 7.1. Both speech codecs define the coding and decoding process not only for linearly quantized speech data but also for A-law-encoded data. We applied this direct conversion here to minimize rounding errors during conversion.

With the error free transmission model from section 7.2.1 this results in two additional recording sets that we denote shortly as ‘sim G FR’ for the simulated FR recordings and as ‘sim G EFR’ for the EFR codec respectively. The feature vectors based on the transcoded data are computed in the same manner compared to the unprocessed speech data.

When regarding the channel model including transmission errors from section 7.2.2 we receive two data sets for the corresponding error parameters named *sim G FR C-0.08* and *sim G FR C-0.13*. The speech codec is kept fixed here with the FR speech codec.

The general setup for generating the transcoded corpora is shown in figure 7.3.

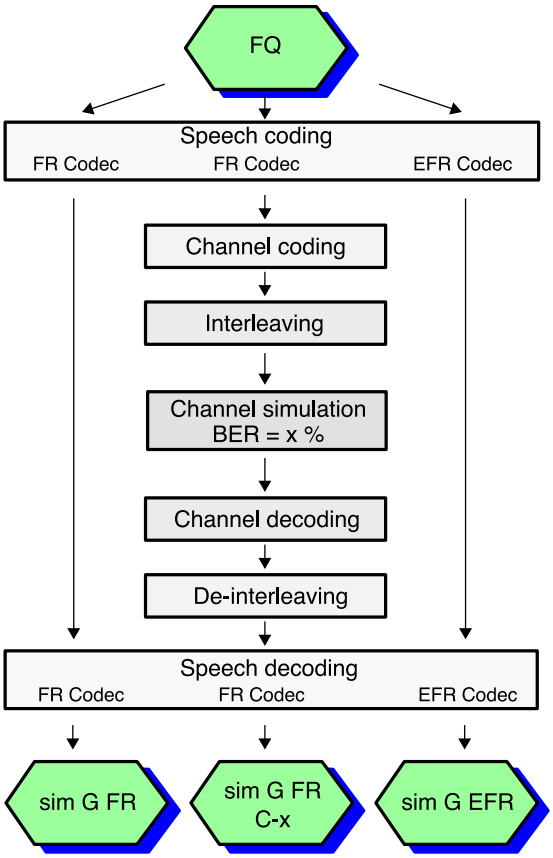

Figure 7.3: Generation of transcoded corpora from the sub-part of FQ recordings of the VeriDat corpus. The simpler approaches use a concatenation of the speech coder and speech decoder (branches to the left and to the right). The branch in the middle simulates transmission errors by a simulated channel with selectable bit error rate.

As we have already stated this simulation technique can reproduce only a part of the real acoustic effects in the GSM recordings. First, the GSM simulation uses already A-law compressed data while in real applications the GSM coding is performed using linearly quantized data obtained directly from the analog microphone signal. Moreover the characteristics of the microphones used in both types of handsets differ and the geometrical positioning of the microphone towards the speaker changes due to differences in the handset shapes.

In addition, the channel effect of the analog transmitting part of the fixed telephone line is contained in the simulated GSM recording as well. Distortions due to various filtering and noise are not comparable to the GSM channel specific effects that are mainly dominated by transmission errors and interpolated data. Even the transcoded data sets that regard GSM transmission errors still suffer from the inherently existing effects of the fixed line channel.

Finally, genuine GSM recordings normally take place in acoustically different surroundings compared to fixed line recordings. Generally we can assume a higher level of background noise especially for outdoor recordings and resulting speaker effects due this environment, namely the Lombard effect.

However, transcoded data sets from fixed line data has already been used for various channel adaptation experiments in speech recognition [62, 63]. In section 8.4 we will present in more detail related experiments from literature within the speaker verification domain.

7.4 Impact of channel effects in the cepstral domain

Several major environmental effects on speech represented as predictor coefficients or LP-based cepstral coefficients are investigated in [64] and [65]. We will follow the notation introduced there and present the results for the cepstral domain of the two main effects, namely distortion by linear channel and distortion by additive noise.

Linear channel

Following [65, formula 7], the effect of a linear channel given by the impulse response $h(n)$ is described by the convolution operation

$$s'(n) = s(n) \star h(n), \tag{7.3}$$

where $s'(n)$ denotes the distorted speech signal and $s(n)$ denotes the clean speech signal. The predictor coefficients of the distorted speech, calculated with the autocorrelation method, are given by

$$\begin{aligned} \mathbf{a}' &= \mathbf{R}_{s'}^{-1}\mathbf{r}_{s'} && (7.4)\\ &= s(0)(\mathbf{HS})^{-1}\mathbf{h}_1 + \mathbf{a} && (7.5) \end{aligned}$$

where the matrix $\mathbf{H}$ can be derived from the impulse response $h(n)$, the matrix $\mathbf{S}$ depends on the input speech $s(n)$ and $\mathbf{h}_1$ is a column vector of the impulse response values $[h(1), h(2), \cdots, h(N)]^T$. Clearly, the new vector of the predictor coefficients is a translated version of the original vector where the translation vector depends on the current speech signal and the characteristics of the channel. .

Additive Noise

Additive noise in the speech signal is modeled in our case as white noise. The distorted speech signal is given by

$$s'(n) = s(n) + q(n) \tag{7.6}$$

where the noise $q(n)$ has zero mean and constant power σ^2:

$$E\left[q(n)\right] = 0 \quad \text{and} \quad E\left[q^2(n)\right] = \sigma^2. \tag{7.7}$$

The prediction coefficients of the noisy speech $\mathbf{a}'$ are calculated as ([65, formula 3])

$$\mathbf{a}' = (\mathbf{R}_s + \sigma^2\mathbf{I})^{-1}\mathbf{R}_s\mathbf{a}, \tag{7.8}$$

where $\mathbf{R}_s$ denotes the autocorrelation matrix. A singular value decomposition as shown in [64, formula 38] unveils that the norm of the predictor coefficients is reduced when speech is corrupted by additive white noise. The general orientation of the predictor coefficient vector is maintained, however. Please note that no translation of the predictor coefficients takes place.

Effects in the LP-based cepstrum

As we have previously presented at the end of section 3.2, the cepstrum based on the LP-spectrum of time-domain signal is given by

$$c_{lp}(n) = \mathcal{Z}^{-1}\left[\log S(z)\right] = \mathcal{Z}^{-1}\left[\log \frac{1}{1-\sum_{i=1}^{p} a_i z^{-i}}\right]. \tag{7.9}$$

There is a indirect recursive method [20] and several direct methods [66] to compute the cepstrum coefficients using the predictor coefficients a_i. In [65, formula 9], another direct computation method is given:

$$c_{lp}(n) = \frac{1}{n}\sum_{i=1}^{p} i\, a_i\, h_s(n-i), \tag{7.10}$$

where $h_s(n)$ denotes the impulse response of the inverse filter of $A(z)$. Written in matrix form, the cepstral coefficient vector $\mathbf{c}_{lp}$ is calculated from the predictor coefficient vector $\mathbf{a}$ as

$$\mathbf{c}_{lp} = \mathbf{D}_1\mathbf{H}_s\mathbf{D}_2\mathbf{a}, \tag{7.11}$$

where $\mathbf{D}_1$ and $\mathbf{D}_2$ are diagonal matrices and $\mathbf{H}_s$ depends on the impulse response (see [65, formula 10] for more details).

Linear channel

In case of a linear channel, we receive the cepstral coefficients of the filtered speech as

$$\mathbf{c}'_{lp} = \mathbf{D}_1\mathbf{H}\mathbf{D}_1^{-1}\mathbf{c}_{lp} + s(0)\mathbf{D}_1\mathbf{H}\mathbf{H}_s\mathbf{D}_2(\mathbf{H}\mathbf{S})^{-1}\mathbf{h}_1. \tag{7.12}$$

In addition to the translation of the cepstral coefficient we find also a scaling matrix.

Additive white noise

When speech is degraded by additive white noise, the resulting cepstral coefficients are given as

$$\mathbf{c}'_{lp} = \mathbf{D}_1\mathbf{H}_s\mathbf{D}_2(\mathbf{R}_s + \sigma^2\mathbf{I})^{-1}\mathbf{R}_s(\mathbf{D}_1\mathbf{H}_s\mathbf{D}_2)^{-1}\mathbf{c}_{lp}. \tag{7.13}$$

Here no translation takes place but a sound-dependent transformation matrix is applied to the original cepstral vector.

Multiple effects

Due to the transitivity of the affine transformation, a sequence of distortions, that can be described by affine transformations, result again in an affine transformation:

$$\mathbf{c}_2 = \mathbf{A}_1\mathbf{c}_1 + \mathbf{b}_1 \quad \text{and} \quad \mathbf{c}_3 = \mathbf{A}_2\mathbf{c}_2 + \mathbf{b}_2 \tag{7.14}$$

results in

$$\begin{aligned} \mathbf{c}_3 &= \mathbf{A}_3\mathbf{c}_1 + \mathbf{b}_3 \quad \text{with} && (7.15)\\ \mathbf{A}_3 &= \mathbf{A}_2\mathbf{A}_1, && (7.16)\\ \mathbf{b}_3 &= \mathbf{A}_2\mathbf{b}_1 + \mathbf{b}_2. && (7.17) \end{aligned}$$

Thus, the effects of a sequence of noise and channel interference can still be described by an affine transformation.

Expected effects in GSM coded speech

We expect to find both mentioned effects in the LPCC domain of GSM coded speech: scaling of the feature vectors due to additive noise and translation due to linear filtering. Although the GSM coding/decoding sequence must be regarded as a highly non-linear filter we might expect that for similar sections of a speech signal (i.e. similar phonemes or similar classes of phonemes) the coding will introduce consistent effects in the cepstral domain. We will investigate in the following section if we can verify this hypothesis. In that case we could assume that the speech coder shows a piece-wise linear behaviour as a simple approximation for similar sections in the speech signal.

7.5 Effects on broad phonetic classes in the LPCC domain

In order to investigate the effect of the GSM codec in the cepstral domain more detailed, a small sub-corpus with manually segmented recordings was derived from the VeriDat database. The phonetic segmentation information serves as a basis for a more coarse segmentation which uses broad phonetic groups e.g. vowels or fricatives. The cepstral vectors of the sub-corpus are clustered based on these broad phonetic groups. Clusters from the FixedQuiet recordings are then compared with their corresponding clusters in either transcoded FixedQuiet recordings (i.e. simulated GSM recordings) or real GSMQuiet recordings.

The measurements are presented both qualitatively and quantitatively: scatter plots depict the effects of GSM coding for selected pairs of cepstral coefficients while distance measurements between uncoded and GSM-coded clusters are calculated using the Bhattacharyya distance.

Our primary interest is to enhance the robustness of LPC based cepstrum (LPCC). Thus all cepstral vectors are obtained of the LPCC_SD parametrisation described in section 3.2.

7.5.1 Manually segmented sub-corpus

The main purpose of the sub-corpus is to provide a high quality phonetic segmentation from 30 different speakers with linguistic similar content. We selected therefore one of the fixed items in the VeriDat database, item F1 containing the German version of the sentence "My voice is my password". For 28 out of 30 speakers, a F1 item from a single session out of in total 20 sessions was randomly selected while for speaker 0048 (female) and 0049 (male) one recording from each environment type (FixedQuiet, FixedNoisy, GSMQuiet, GSMNoisy) was randomly drawn. Due to the random selection process the frequencies on the environment are not balanced: there are 15 FQ, 9 GQ, 9 FN and 3 GN utterances. More detailed information about the selected sessions is provided in appendix F.

The recordings have been segmented by a phonetically trained expert using the labeling convention of the BITS project [67]. The label set is based on the German SAMPA system[2] and is extended in some aspects: the silence and the burst phase of plosives is labeled separately and additional smacks are denoted by the character '§'. The complete set of all occurred phonemes is given in table 7.1. In addition the table shows the clustering of the phonemes to broad phonetic groups such as vowels, fricatives, plosives, approximates, nasals and pauses.

[2]See http://www.phon.ucl.ac.uk/home/sampa/german.htm

vowels	fricatives	plosives	nasals	approximates	pauses
/@/, /E/, /I/, /Iq/, /O/, /O6/, /O6§/, /a/, /aI/, /6/, /_6/	/S/, /s/	/k_b/, /k_s/, /k_s§/, /t_b/, /t_s/,/t_b§/, /t_s§/, /Q_b/, /Q_s/, /k/, /t/, /Q/	/m/, /n/, /n§/	/R/, /v/, /v§/	/<p:>/, /<br:>/, /_p:/, /_nib/, /_usb/

Table 7.1: Phonemes occurring in manually segmented F1-corpus, notation according to [67], phonemes grouped in broad phonetic groups

7.5.2 Clustering of cepstral vectors

The temporal information from the manual segmentation is used to cluster the LPC-based cepstral vectors from the original recordings and the transcoded recording sets "sim G FR" and "sim G EFR". As we stated already before, we restrict our experiments to the quiet recordings and denote these subsets as "sim GQ FR" and "sim GQ EFR".

The time resolution of the sample based segmentation is much higher than the time resolution of the 25 ms wide frames from the feature extraction stage. In case that more than one segment falls into a frame, the frame is assigned to the phonetic class covering most of the frame duration.

7.5.3 Recording sets

There are three GSM recording sets ("G","sim G FR" and "sim G EFR") whose cepstral feature vectors can be compared with the fixed line recordings ("F"). As we stated above, our analysis is restricted to the quiet utterances in order to exclude additional effects of loud background noises:

- FQ vs. GQ. The recordings are taken from the genuine VeriDat recordings, no transcoding is applied.
- FQ vs. sim GQ FR. The simulated GSM recordings are created by passing the FQ utterances through a coder-encoder chain of the GSM FR speech codec (see 7.3).
- FQ vs. sim GQ EFR. Similar to the previous point, except that the GSM EFR speech codec is used.

Both comparisons using the transcoded data will reveal the sole effects of the speech codecs. However the comparison using the genuine GSM utterances suffers from additional variability as different recording sessions are involved. By selecting items with fixed lexical content (F1 item) and moderate background noise (quiet recording subset) we hope that the results for the comparison type 'FQ vs. GQ' are still comparable to those using the transcoded recordings.

7.5.4 Scatter plots

Our first approach to gain insight into the effect of GSM coding in the cepstral domain uses scatter plots of pairs of cepstral coefficients. Figure 7.4 shows the cepstral clusters of the four highest populated phonetic groups from speaker 0049: vowels, fricatives, plosives and nasals. In the upper row a single FQ utterance is used while in the lower row a single genuine GQ utterance is depicted.

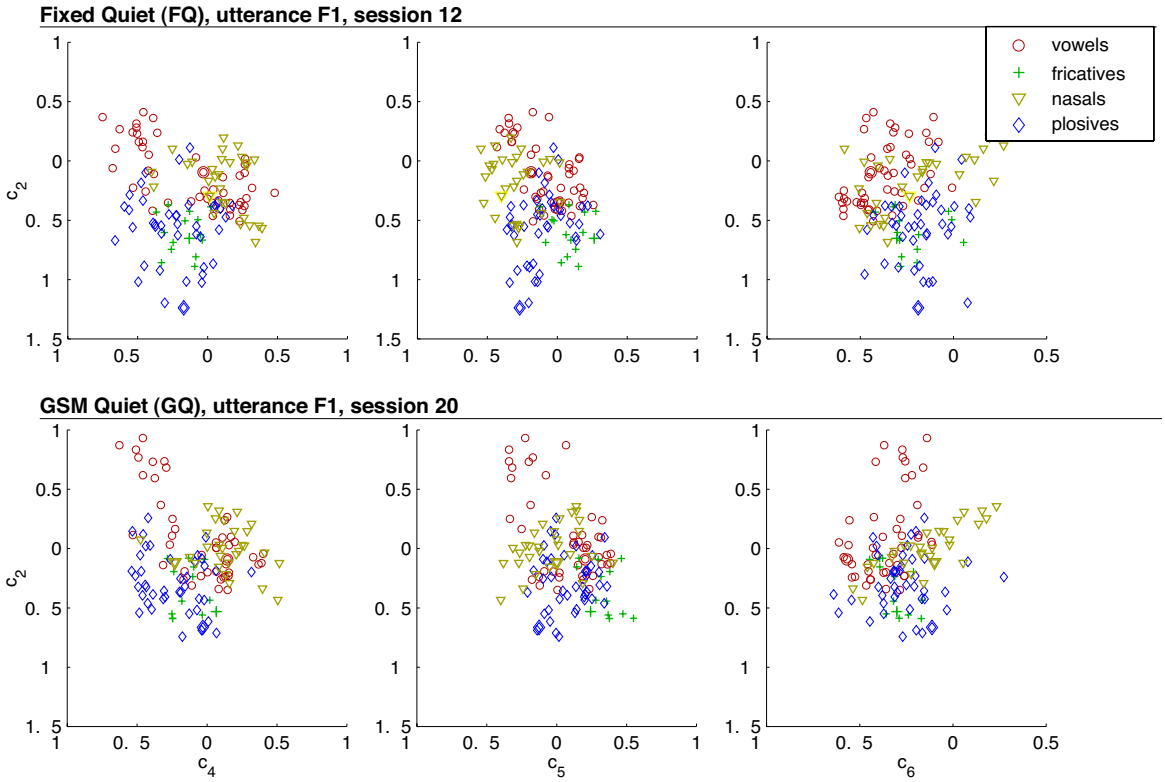

Figure 7.4: Scatter plots of cepstral feature c_2 versus c_4 (left), c_5 (middle) and c_6 (right). Single recording session of item F1 taken from speaker 0049. Top row shows a recording from the 'FixedQuiet' data, bottom row a recording from the 'GSMQuiet' data.

Plotting the second cepstral coefficient, c_2, against c_4, c_5 and c_6 should exemplify the effects while focusing on only a few cepstral pairs. The plots are difficult to interpret in case that no color information is available. However a few basic effects can be seen. The vowels are affected most in c_2 as they are stretched nearly up to the value $+1$. The remaining cepstral coefficients c_4, c_5 and c_6 are far less affected. The group of fricatives and nasals in turn seems to be most affected in the three mentioned cepstral components.

The scatter plots are each based on data from a single utterance. Table 7.2 shows the population of the phonetic classes for the case of a single speaker, speaker 0049 (also used in the scatter plot from figure 7.4), and for the case of pooling all segmented data from all speakers in the sub-corpus. The cluster population for the FQ recordings and the transcoded GSM recordings are the same as the segmentation information can also be applied to all recordings derived from the original FQ recordings. The genuine GQ recordings show a slightly different cluster population. Because the approximates are rarely populated we discard this phonetic group from our further investigations and continue with four phonetic groups: vowels, fricatives, plosives and nasals.

The scatter plots can not give the whole picture of the complete shifts and transformations of the cepstral clusters resulting from GSM coding. We will quantify the effects using the Bhattacharyya distance in the following. Before that we will briefly present the properties and motivation of the Bhattacharyya distance measure.

Recording type	No. of F1 items	Vowels	Fricatives	Plosives	Nasals	Approximates
Single speaker (0049)						
'FQ' (also 'sim GQ FR', 'sim GQ EFR')	1	49	15	41	29	5
'GQ'	1	45	13	42	29	5
Pooled data from sub-corpus						
'FQ' (also 'sim GQ FR', 'sim GQ EFR')	15	1108	246	646	613	83
'GQ'	9	652	159	360	362	43

Table 7.2: Properties of the 'FixedQuiet' speech data derived from a single speaker (0049) and from the complete data set (pooled from all speakers): number of recording items and number of feature vectors grouped by phonetic classes.

7.5.5 The Bhattacharyya distance measure

In statistics, several proximity degrees between two different probability densities have been developed. Most of them can be regarded as a dissimilarity measure and can therefore be related with the notion of a distance measure i.e. more dissimilar probability densities are characterised by a greater distance. The Bhattacharyya distance b of two probability densities $p_1(x)$ and $p_2(x)$, resulting from two different classes of vectors, is given by [68]

$$b = -\ln \int_{-\infty}^{+\infty} \sqrt{p_1(x)p_2(x)}\, dx \tag{7.18}$$

In case that the probability densities $p_1(x)$ and $p_2(x)$ can be modelled by multivariate Gaussian distributions, a special case of the Bhattacharyya distance can be calculated explicitly [69]. If we consider μ_i as the mean vector and Σ_i as the covariance matrix of the density $p_i(x)$, we can write

$$b = b_m + b_c \tag{7.19}$$

$$= \frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1}(\mu_1 - \mu_2) + \frac{1}{2}\ln\left(\frac{\det\left(\frac{\Sigma_1+\Sigma_2}{2}\right)}{\sqrt{\det(\Sigma_1)\det(\Sigma_2)}}\right). \tag{7.20}$$

The Bhattacharyya decomposes therefore into two parts. The first one is based solely on the difference in the means (b_m) scaled according a mean covariance matrix. It describes the distance based on simple translations of the clusters. The second part, denoted as b_c, captures changes of the clusters' form like e.g. rotations or scaling.

Although the Bhattacharyya distance is not a real metric since it does not obey the triangle inequality, it is popular for classification problems. Using its closed form solution for Gaussian mixture requires modelling of an empirical distribution by an unimodal multivariate Gaussian distribution. This two-step procedure might introduce errors due to the additional modelling. Alternatively, one could use a nonparametric approach [70] that computes the Bhattacharyya distance directly from the data samples. In our case we apply the first approach by estimating a single Gaussian mixture.

Since we use diagonal covariance matrices, the Bhattacharyya distance of a p-dimensional Gaussian distribution can be written as a sum of p Bhattacharyya distances based on one-dimensional

distributions:

$$b = \sum_{i=1}^{p} b_i \quad \text{with} \tag{7.21}$$

$$b_i = \frac{1}{4}(\mu_{1i} - \mu_{2i})^2 \left(\frac{1}{\sigma_{1i} + \sigma_{2i}}\right) + \frac{1}{2}\ln\left(\frac{\left(\frac{\sigma_{1i}+\sigma_{2i}}{2}\right)}{\sqrt{\sigma_{1i}\sigma_{2i}}}\right). \tag{7.22}$$

For increasing dimensionality p, the simplified Bhattacharyya distance is monotonically increasing because the distance contribution of each added dimension is greater or equal to zero.

Note that a cluster modelling using only diagonal covariance matrices is less detailed than using a cluster modelling with full covariance matrices. Some kind of transformations e.g. a rotation of clusters around their mean are not reflected properly by the simplified Bhattacharyya distance. However the simpler modelling is appropriate for the small cluster population from which the model parameters have to be estimated. The group of fricatives shows e.g. for speaker 0001 a population count as low as 13.

Only in the case of the pooled data the population counts would be large enough to estimate full covariance matrices. Bhattacharyya distances measured on these elaborated Gaussian models are not comparable to those measured on the Gaussian models using diagonal covariance matrices. We decided therefore to continue calculating the Bhattacharyya distance based on the low complexity (i.e. single-mixture) Gaussian models.

Please note that modelling the clusters with a single Gaussian introduces additional errors in estimating the distance between clusters. However this remains the only way to compute Bhattacharyya distances in a feasible way.

7.5.6 Bhattacharyya distance between clusters

The cepstral clusters from the four different phonetic groups are modelled by unimodal multivariate Gaussian probability distributions using diagonal covariance matrices. In the result reported here, only the first twelve cepstral coefficients are included. As shown in formula 7.22 the computation of the Bhattacharyya distance can be simplified by calculating the Bhattacharyya distance of each feature pair independently.

Figure 7.5 shows the Bhattacharyya distance b and its mean part b_m between the clusters from the FQ and the GQ data. Again, the comparisons were made using speech data from speaker 0049. The largest distances occur for the clusters of the fricatives and the nasals while the vowels and plosives are less affected. The comparison involving the genuine GSM recording shows a larger distance to the fixed line quality recording than the comparison using the simulated data. The features of the two different transcoded recording sets, the “sim GQ FR” and the “sim GQ EFR”, show a much smaller distance to the FQ data. In addition the distances are roughly constant, regardless of the phonetic group.

While comparing b and b_m it seems that the differences in the cepstral domain between the fixed line data (FQ) and genuine GSM recordings (GQ) are mainly based on simple translations of the clusters: the Bhattacharyya distance is dominated by the mean-based component. The distances with the transcoded data involved are composed by both components of the Bhattacharyya distance: b_m and b_c are in comparable order of magnitude. This indicates that also a scaling of the variances is involved. The absolute contribution of the second part of the Bhattacharyya distance, b_c, seems to be roughly constant regardless of the recording set.

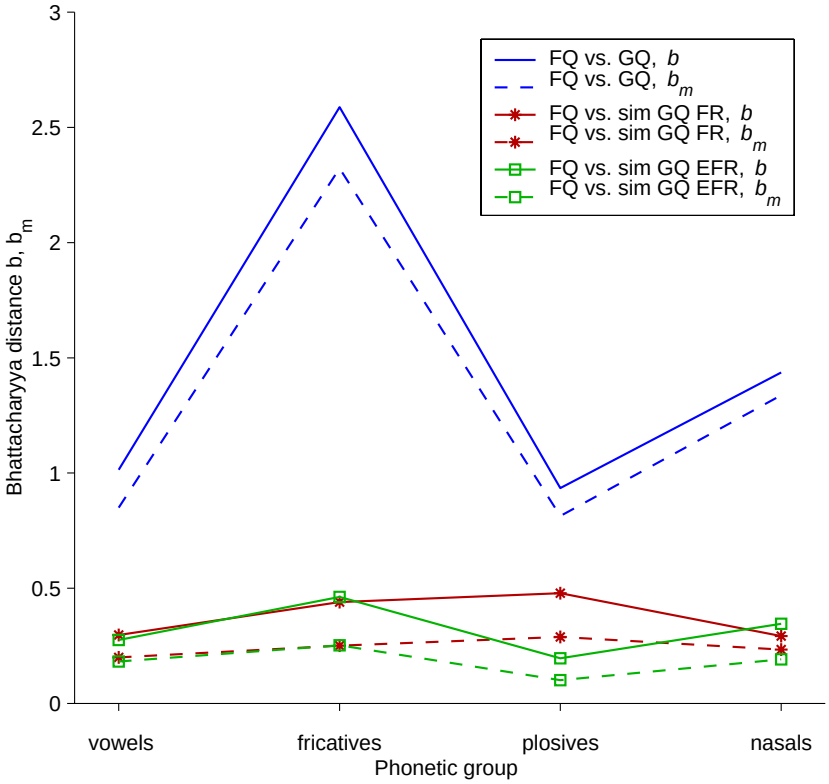

Figure 7.5: Data from speaker 0049 (see upper part of table 7.2): total Bhattacharyya distance and mean part of the Bhattacharyya distance depending on various phonetic groups. The distance is calculated between 'FQ' data and either genuine 'GQ' or simulated 'GQ' data.

Equivalent characteristics of the Bhattacharyya distance can be found for the remaining speakers of the sub-corpus although the absolute level of the Bhattacharyya distance as well as the prominence of the distance peak for the fricatives varies from speaker to speaker.

The same investigation has been done by pooling the data from all speakers in the sub-corpus. Figure 7.6 shows that only for the distance between GQ and FQ the same distance peak for the fricatives can be found as in the single-speaker case. The distances on both GSM codecs show a roughly equivalent characteristic within the same value range observed in the single-speaker comparison. For the GSM FR codec the distances of the phonetic classes plosives and nasals are comparable to those from the GQ recordings. In case of the GSM EFR codec, the distances are clearly the lowest in every phonetic group and are nearly independent from the type of the phonetic group.

In addition, the absolute distance level is much lower because the underlying Gaussian probability distributions cover a wider feature space: incorporating data of several speakers leads to a higher variability. This renders overlapping of the distributions from different phonetic groups more likely.

To summarise the results: we find a prominent distance between FQ and GQ recordings for the fricatives, both in single-speaker comparisons and in a comparison done on the pooled data. A second, less prominent distance peak can be found for the nasals in the data of some speakers. The main part of the differences is described by the mean part of the Bhattacharyya distance.

The distance between the 'FQ' and both types of the 'sim GQ' recordings tends to be lower

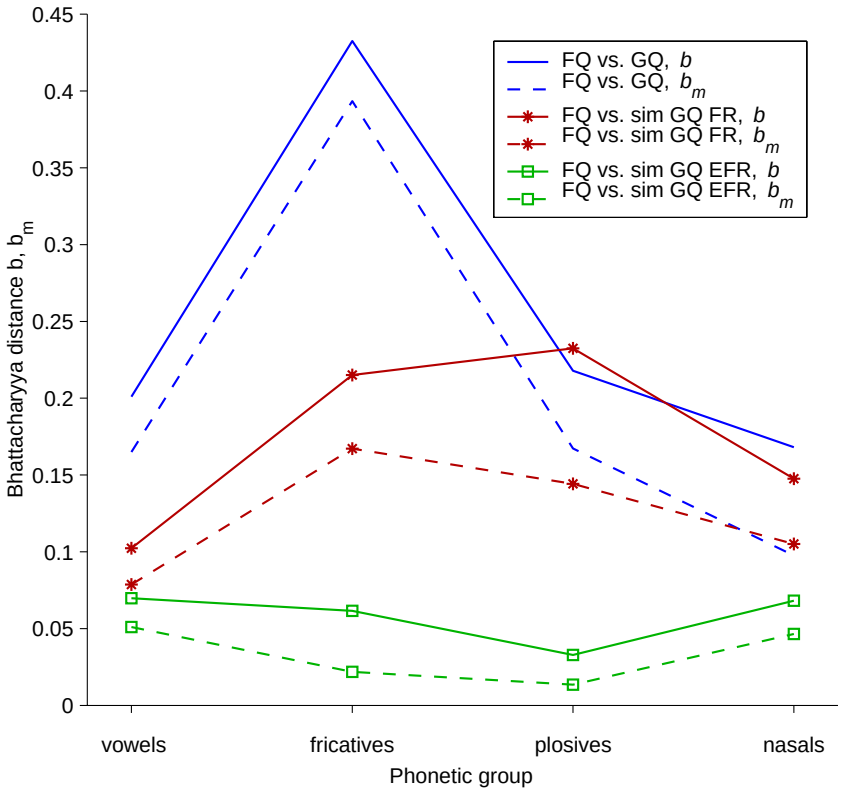

Figure 7.6: Pooled data from all speakers (see lower part of table 7.2): total Bhattacharyya distance and mean part of the Bhattacharyya distance depending on various phonetic groups. The distance is calculated between 'FQ' data and either genuine 'GQ' or simulated 'GQ' data.

than the distance towards the real 'GQ' recordings. The latter seems to be independent from the phonetic class as well.

Although the Gaussian probability distributions estimated on the fricatives data are based on few observations in the single speaker case, we find the same peak of the Bhattacharyya distance in the more populated set using the pooled speaker data. We might argue that the peak is a consistent property of the data and not an artifact due to data sparsity.

We might also conclude that the cepstral clusters in the FQ and GQ recordings are simply translated in the feature space. This could motivate a correction scheme for GSM coded recordings that uses an independent translation vector for different phonetic groups. However the real transformations of the feature space from the FQ to GQ case might be more complex. As we stated above, the simplified version of the Bhattacharyya distance used here does not capture complex transformations of the cepstral clusters e.g. a varying correlation of cepstral coefficients between the fixed line quality and the GSM coded recordings. Therefore, compensation of the variance scaling might give additional benefit for the adaptation process.

7.6 Comparison of simulated GSM data with real GSM data

We have seen in the last section that the simulated GSM recordings do not reach the same distances from the fixed line data compared to the real GSM data. The loss of quality due to the codec type can be seen for the data set averaged over all speakers: the high-quality EFR codec

results in smaller distances than the less sophisticated FR codec. However, the distances found with the real GSM data is only reached for the plosives and the nasals. For individual speakers, we often find that the contribution of the codecs to the total distance measure is as low as 20%.

Like many distance measures that express complex differences by a single figure, we must note that two Bhattacharyya distances with the same value does not necessarily indicate that both data sets result from the same distribution. Thus, even for the plosives and the nasals from the averaged data of all speakers, we can not state that the simulated GSM recordings include the same effects we find in the real GSM recordings.

The basic behaviour of the distances for different phonetic groups might be explained by the nature of the LPC coding used in the GSM codecs. While the voiced phonemes such as vowels and nasals result in a residuum signal that can be easily compressed, the voiceless phonemes such as plosives and fricatives produce a random residuum signal that suffers from the downsampling stage. Therefore we reason that plosives might be less affected since a large part of the speech segment is filled by the occlusion phase that can be easily compressed.

We must conclude that only a part of the effects in real GSM speech can be attributed to the codec itself. Other effects such as the recording environment, transmission quality, handset properties or the influence of speaking habits sum up to the differences we find between the fixed line and the GSM recordings. It seems that for at least some speakers these additional effects are indirectly related to the individual speaker.

Similarly to the contribution of the GSM codec alone we could expect that speaker models adapted with simulated GSM coding can only partly decrease the mismatch to real GSM recordings.

Chapter 8

Adaptation to GSM channel effects

In the previous chapter we found different degrees of the effect of GSM speech coding on LP-based cepstral coefficients, depending on their association to broad phonetic classes. In this chapter we will first outline shortly different possibilities to adapt HMM models and afterwards present a short overview of maximum likelihood based techniques and its variants deployed both in the feature and the model space. We will continue with outlining our proposed adaptation scheme for compensation of GSM coding effects. In order to present it in the context of the ongoing research in this area, we will give an overview of recent adaptation techniques developed in the context of speaker recognition for compensating the effects of the transmission channel. Finally, in the largest part of this chapter, we will focus on our model adaptation scheme based on MLLR in more detail. We will describe different choices of adapted parameters of the HMMs, introduce our implementation of a variance-only scaling scheme and finally present the generation of phonetically motivated regression classes used in the MLLR adaptation system.

8.1 Adaptation methods for HMMs: MLLR and MAP

Reliable adaptation algorithms are one of the key requirements for pattern recognition systems that should perform under varying conditions. For HMM based speech recognition systems many different approaches have been developed to adapt the models to the current speaker or to the current environment. Most of the common techniques can be used for both adaptation purposes. The goal is to obtain the performance level under matching conditions (speaker and environment matching to conditions during model training) with only a small amount of condition-specific (either speaker-specific of environment-specific) data.

In the field of speaker recognition the impact of transducer mismatches is addressed using different adaptation schemes. The best known methods are MAP (maximum a posteriori) training and MLLR (maximum likelihood linear regression) adaptation.

Originally both methods were developed for speaker adaptation for speaker-independent speech recognizers. However the same techniques and related methods have also been applied for environment adaptation in speech recognition [71] and speaker recognition [72, 73].

Often adaptation techniques are distinguished by their mode: either supervised or unsupervised mode may be used. If the characteristics of the adaptation are labeled and thus its category is known, we term it as supervised adaptation. Alternatively, the adaptation data is unlabelled and the system has to determine the missing information. In case of a speaker adaptation of a speech recognizer the missing labelling data comprises the utterance spoken. Here we use the adaptation for an environment adaptation: we supply a labeling of the environment together with

the adaptation data; the linguistic content is always known. Thus we will use only supervised adaptation in the following.

Generally the adaptation methods can be divided into maximum likelihood (ML) transformation-based approaches and Bayesian techniques. The MLLR adaptation method [74] is the most famous member of the former group while the MAP adaptation is the best known member of the latter one. ML transformation-based methods [75, 74, 71] aim at adjusting indirectly the model parameters via a small number of transformations. On the one hand, this limits the capability to capture the fine structure in the transformation process but, on the other hand makes the estimation of the transformation parameters robust and efficient with a limited amount of adaptation data.

In contrast to the ML-based methods, the Bayesian techniques [76, 77] aim at adjusting the model parameters directly. This approach requires a large amount of adaptation data and can therefore be more accurate in capturing fine details of the adaptation data.

In the following we will present briefly the most famous representative for each adaptation approach: the Maximum Likelihood Linear Regression (MLLR) for the ML-based techniques and the Maximum A Posteriori (MAP) for the Bayesian techniques. Later in section 8.2, a more detailed classification of ML-based methods is presented.

Maximum likelihood linear regression (MLLR)

The MLLR adaptation method [78, 74] is more effective with only small amounts of adaptation data. It estimates a set of transformations that can be applied to the model parameters. If a group of model parameters can be assumed to be transformed similarly, a single transformation set can capture general relationships between the old model and the model of the new speaker or environment. The type of transformations is restricted to linear transforms and the parameters are estimated by maximizing the likelihood on the given adaptation data. Later in section 8.5 we will focus on the application of MLLR for adapting speaker models.

Maximum a posteriori training (MAP)

The MAP training ([76, 77, 75], also known as Bayesian adaptation or Bayesian learning) updates the HMM model parameters by joining known information (the old parameters) with the statistics derived from the adaptation data. Like the EM algorithm, the adaptation process decomposes into two steps. The first step is identical to the first step of the EM algorithm where statistics required for computing the mixture weight, mean and variance are collected. In the second step the statistics from adaptation data are combined with the old statistics from the HMM model using a data-dependent weighting coefficient. The data dependency is designed to weight the statistics with high counts of the adaptation data more than low counts of the data. Following an example from [73, section 3.4] the mean parameter μ of a mixture m is based on the statistic

$$E(\mathbf{o}(t))_j = \frac{1}{n_j} \sum_{t=1}^{T} \mathrm{P}(q_m(t)|\mathbf{o}(t), \lambda)\mathbf{o}(t) \tag{8.1}$$

where T is the number of adaptation frames and $n_j = \sum_{t=1}^{T} \mathrm{P}(q_m(t)|\mathbf{o}(t), \lambda)$ is the total count for being in the particular state depending on the given adaptation data. The new mean $\hat{\boldsymbol{\mu}}_j$ is a weighted sum of the old and the new statistic:

$$\hat{\boldsymbol{\mu}}_j = \frac{n_j}{n_j + \rho} E(\mathbf{o}(t))_j + \frac{\rho}{n_j + \rho} \boldsymbol{\mu}_j. \tag{8.2}$$

The data-dependency of the weighting coefficient is realised by the relevance factor ρ. Its value marks the point where the data count of the adaptation data has the same weight as the old mean parameter. Higher values of ρ gives more weight to the prior information i.e. the old parameters.

The drawback of the MAP adaptation method is that it is an unconstrained method and updates therefore only those parameters for which observations occur. It requires a relatively large amount of adaptation data in order to be effective for also sparsely occupied Gaussian mixtures.

8.2 Categorising ML-based adaptation methods

In general, ML-based adaptation methods allow the computation for parametrised adaptation for any type of transformation functions. Depending on the domain where the transformation takes place, these techniques can be further divided into feature transformation methods and model transformation methods (see e.g. [79]).

Feature-based transformations

Feature-based approaches attempt to modify the distorted features and transform them into the original (undistorted) features. Thus the transformed features fit the clean speech or speaker models better.

Models using multiple Gaussian mixtures describe the pdf of the observations as

$$b_j(\mathbf{o}(t)) = \sum_{m=1}^{M} w_{jm} \mathcal{N}(\mathbf{o}(t), \boldsymbol{\mu}_{jm}, \boldsymbol{\Sigma}_{jm}) \tag{8.3}$$

Here, we denote with $\mathbf{o}(t)$ the unadapted features. The adaptation in the feature space hypothesizes a transformation $\hat{\mathbf{o}}(t) = f_\nu(\mathbf{o}(t))$ where ν denotes the transformation parameters that have to be estimated. In addition we denote the resulting change of the Gaussian parameters in the feature domain which correspond to the changes in the model domain. In order to simplify the notation scheme we present only the results for diagonal variance matrices i.e. we do not take into account the covariation between individual vector components. The original Gaussian parameters for a single mixture are denoted as $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ while the parameters after the transformation are described by $\hat{\mu}$ and $\hat{\sigma}$.

Various different transformations are used:

- Diagonal Affine Transform [75]

  $$f_\nu : \quad \hat{\mathbf{o}}(t) = \boldsymbol{A}\mathbf{o}(t) + \boldsymbol{b} \tag{8.4}$$

  This results in an observation density

  $$b_j(\hat{\mathbf{o}}(t)) = \sum_{m=1}^{M} w_{jm}\, \mathcal{N}(\mathbf{o}(t), \boldsymbol{A}\boldsymbol{\mu}_{jm} + \boldsymbol{b}_m, \boldsymbol{A}\boldsymbol{\Sigma}_{jm}\boldsymbol{A}^T) \tag{8.5}$$

  Typically a diagonal matrix $\boldsymbol{A} = \mathrm{diag}(\sigma_1^2, \ldots, \sigma_n^2)$ is used.

Under this constraint the typical transformation of the statistical parameters of a single component i of a Gaussian mixture is given as:

$$\hat{\mu}_i = a_{ii}\,\mu_i + b_i, \tag{8.6}$$
$$\hat{\sigma}_i^2 = a_{ii}^2\,\sigma_i^2. \tag{8.7}$$

- Stochastic matching [80, 71]
  Originally, stochastic matching (often also labeled as stochastic feature transform, SFT) was designed for speech recognition in order to match a test utterance from a distorted environment to models trained from clean data. The match can either be performed in the feature domain or by transforming the model parameters (similar to MLLR). Its unique feature is that it operates on a single test utterance. The transformation parameters are estimated by the EM algorithm. Typically, in speech recognition a joint maximisation takes place when iteratively maximizing the word sequence W while keeping the transformation parameters ν constant and then maximizing the parameters ν by keeping W constant (see [71, section II]).

  However, the application of stochastic matching in speaker verification requires the use of an additional training set; we present the reason later in section 8.2. Thus, all applications of stochastic matching for speaker models like e.g. [81, 82], operate on extra training data for estimating the transformation parameters.

  Typically, a transformation function f_ν is defined:

$$f_\nu : \quad \hat{\mathbf{o}}(t) = \mathbf{o}(t) + \boldsymbol{b}_g(\boldsymbol{\mu}_b, \boldsymbol{\sigma}_b) \tag{8.8}$$

  $\boldsymbol{b}_g$ is a vector Gaussian random variable defined by the means vector $\boldsymbol{\mu}_b$ and variances vector $\boldsymbol{\sigma}_b$. The resulting transformation of the parameters of a single Gaussian mixture model is given as:

$$\hat{\mu}_i = \mu_i + \mu_{b_i}, \tag{8.9}$$
$$\hat{\sigma}_i^2 = \sigma_i^2 + \sigma_{b_i}^2. \tag{8.10}$$

  When using a deterministic bias shift ($\sigma_b = 0$), the features are translated in the feature space by $\boldsymbol{b}$:

$$f_\nu : \quad \hat{\mathbf{o}}(t) = \mathbf{o}(t) + \boldsymbol{b} \tag{8.11}$$

  As shown in [83] the reestimation of the vector $\mathbf{b}$ via the EM-algorithm is done by

$$\boldsymbol{b}' = \frac{\sum_{t=1}^{T}\sum_{j=1}^{M} h_j(\hat{\mathbf{o}}(t))\,(\boldsymbol{\mu}_j - \mathbf{o}(t))\boldsymbol{\Sigma}_j{}^{-1}}{\sum_{t=1}^{T}\sum_{j=1}^{M} h_j(\hat{\mathbf{o}}(t))\,\boldsymbol{\Sigma}_j{}^{-1}} \tag{8.12}$$

  where $\boldsymbol{b}'$ denotes the reestimated bias vector in the iterative EM-algorithm and h_j represents the a posteriori probability of occupying mixture j in the model λ

$$h_j = P(j|\lambda, \hat{\mathbf{o}}(t)) \tag{8.13}$$
$$= \frac{\omega_j\, p(\hat{\mathbf{o}}(t)|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{l=1}^{M} \omega_l\, p(\hat{\mathbf{o}}(t)|\boldsymbol{\mu}_l, \boldsymbol{\Sigma}_l)} \tag{8.14}$$

  and the model λ is composed by a mixture of M Gaussians

$$\lambda = \{\omega_j, \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j\}_{j=1}^{M} \tag{8.15}$$

Both variants differ in the ability to transform either mean or variance of the features independently. While the Diagonal Affine Transform shows a direct connection between the scaling of the original mean and variance the Stochastic Additive Transform allows an independent translation of both parameters.

The feature based approaches using the simple transformations, like the ones we presented above, rely on the assumption that various effects (channel effects, environmental effects) can be approximated by linear filters. However, most distortions are highly non-linear. Even variation of a single component in the transmission chain like e.g. the telephone handset, can exhibit energy-dependent frequency responses where a linear filter may be a poor approximation (see [84] for an example). The same is true for the non-linear effects due to speech coding or transcoding, as we have presented in chapter 7.

Model-based transformations

In contrast to the feature-based transformations the model-based approach uses a transformation $\hat{\lambda} = g_\eta(\lambda)$ where the new models $\hat{\lambda}$ are obtained from the original (clean) models λ. Again the index η of the transformation function indicates that a parameter set has to be estimated to describe the transformation. With the model-based approach the assumption is that the transformed models fit distorted speech (affected by channel effects, environmental effects) much better. Here we focus on models using Gaussian density functions; the original parameters are denoted as $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ while the adapted models are described by $\hat{\boldsymbol{\mu}}$ and $\hat{\boldsymbol{\Sigma}}$.

Commonly deployed variants of model-based transformations include:

- Full Affine Transformation [78]
  This is the founding of the later called approach Maximum Likelihood Linear Regression (MLLR) approach. It is an alternative to the Diagonal Affine Transformation from section 8.2 where only the means of the Gaussian density functions are affected:

$$\hat{\boldsymbol{\mu}} = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}, \tag{8.16}$$

$$\hat{\boldsymbol{\Sigma}}^2 = \boldsymbol{\Sigma}^2. \tag{8.17}$$

  Again the choice between a diagonal transformation and a full affine transformation is possible. Different variants of this transformation are presented in section 8.5.

- Structured Affine Transformation
  In addition to the previous approach the structure of the feature vector with its augmented parts including first and second derivatives can be regarded. Non-zero values for the transformation matrix $\boldsymbol{A}$ may be used for elements whose row and column corresponds to the same feature vector part. For the typically used cepstral feature vectors with appended first and second derivatives, a transformation matrix consisting of three non-zero quadratic blocks is achieved. This avoids ill-conditioned matrix inversions because possibly unrelated correlations between the basic cepstral values and its derivatives are dismissed.

- Scaled Variance Transform [78]
  Here a transformation of the Gaussian means using a bias vector and a scaling factor is applied to each of the Gaussian variances. This technique will be presented in more detail in section 8.5.1.

Differences between feature based and model based transformations

As we have already seen, inherent differences exist in the possible transformations of the Gaussian parameters $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ when linear transformation equations are applied. In addition, different concepts are underlying when non-linear transformations are approximated by piecewise linear transformations.

Both the feature based and the model based transformation technique allow to transform separately different sections of the feature and the model parameters respectively. For the feature-based approaches the feature vectors can be divided by criteria defined by the vector itself. E.g. the codeword-dependent cepstral normalisation (CDCN, see [85]) uses codeword-dependent cepstral biases. This technique allows handling of both channel distortion and background noise, however, it only works well when the level of the background noise is low.

A further distinction results from the fact that both types of adaptation techniques allow pooling of training and adaptation data on different levels. Feature-based techniques can deploy data pools on the acoustical level, based on similarity in this domain. Model-based transformation can group models or parts of models e.g.by selecting individual mixtures in case of HMM based modelling. A common example is the usage of regression trees by MLLR. We will present it later in more detail.

Adaptation of speaker models vs. adaptation in speech recognition

Most of the techniques presented here have their origin in the research of adaptation for robust speech recognition systems. They are often used in an unsupervised adaptation to cancel environment or channel effects. A joint maximization over both the most probable word sequence and the parameters of the adaptation is applied [71]. The maximization over the word sequence serves as constraint for finding the optimum transformation parameters. Therefore the utterance to be decoded can be used directly to find the transformation set.

In speaker recognition our goal is to find a binary decision about the given identity of the claimant. Here, any adaptation technique using the claimant's utterance suffers from the fact, that the speaker model incorporates both the speaker identity and the environment alternatively channel characteristics used during enrolment. Therefore, it is not appropriate to transform either the utterance to the claimed speaker model or to the background model. The first case would possibly adapt impostor speech toward a client model and will result in an increased false acceptance rate, while the second case will lead to an excessive false rejection rate. We will present later in sections 8.4.2 and 8.4.3 a concept by Yiu et al. ([86, 87] to circumvent this problem.

8.3 Proposed adaptation scheme for compensating GSM coding effects

We have seen previously in section 6.3 that the robustness of the LPCC features on the noise level is already high. Therefore we focus our primary interest to ameliorate their robustness to variations of the transmission channel as well.

We propose an adaptation scheme of the speaker models that is based on the well-known MLLR approach. However, the core concept of the scheme is not restricted to the application of the MLLR adaptation.

Our adaptation scheme is intended for a speaker verification system using single-session enrolment by means of a call from a specific telephone network. The goal is to adapt the speaker models to the situation when the client speaker has to be verified from a call originating from a different telephone network. We focus on the situation where the client speaker enrols from a fixed telephone network and tries to authenticate from the GSM network. Thus the client models trained on the fixed telephone data must be adapted to the unseen properties of GSM speech data.

There are two possible sources for the data used in the adaptation process: either genuine speaker data from the application phase of the SV system could be collected or offline data can be applied. Collecting additional client data from the application sessions as adaptation material is error-prune since even with low-FAR thresholds there is a chance of incorporating impostor data into the client model. Therefore the adaptation process should not deploy client utterances from previous accesses. Since we restrict ourselves to single-session enrolment, we do not have the possibility to gather supplementary adaptation data of the client.

Hence we decide to apply offline data which can originate either from a different speaker population or from transcoded client enrolment material.

Similar to the cheating experiments from section 6.4, we state that the SV system receives information about the telephone network used for a call. Based on this knowledge, the system retrieves the matching client model and performs the verification process.

The main requirements of the adaptation system are:

- Single-session enrolment.
- No additional client data for adaptation.
- No adaptation material taken from application phase of SV system.
- Model adaptation shall be performed offline (right after model training).
- Adaptation possible with small amount of adaptation data.
- Information about network type (fixed line / GSM) is available.
- Option to include phoneme-dependent adaptation.

8.3.1 Basic concept

Our adaptation scheme deploys MLLR as a basic adaptation technique since it fits perfectly for the requirements mentioned above. Compared to MAP, MLLR allows better adaptation performance when used with small amounts of adaptation data. Further, the usage of regression trees enables applying different transformations to several model parts. The composition of the regression trees can be based on any criteria. Thus, along with a trivial single regression group comprising all models, individual regression groups i.e. also phonetically motivated groups can be deployed. In addition, some of the used MLLR adaptation schemes are already available in the HTK framework so only modest extensions to the adaptation code of HTK have to be made.

We restrict our experiments to the situation that the enrolment takes place using fixed line data and the intended use of the SV system is over the GSM network. In the following three pictures we depict the usage of different data sets in two different adaptation systems. Common to all three cases is the client model training (which is based on ‘FQ’ data) and the evaluation of the

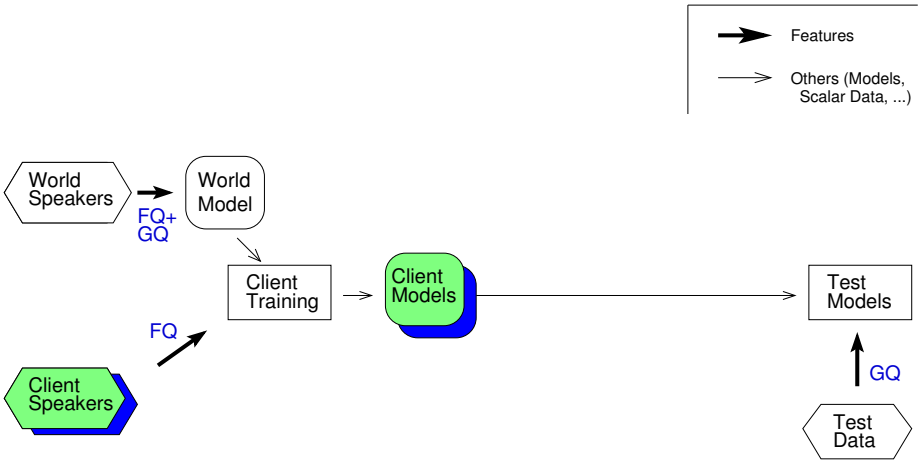

Figure 8.1: Block diagram of the SV system without adaptation scheme. Mismatch between training of speaker models (using 'FQ' data) and evaluation (using 'GQ' data).

SV system using 'GQ' data. The world speakers provide data from both types of networks ('Q' data) to build the world model which in turn seeds the training of the client models.

Figure 8.1 presents the case when no adaptation of the client models is performed. There is a clear mismatch between the training data of the models and the evaluation data. Although the world model with its richer data modeling is used for seeding the client model, the Baum-Welch reestimation process of the client parameters focuses only on the client's training material.

The performance in the mismatched test case will be used as a lower performance bound for adaptation schemes (i.e. an upper bound in terms for the EER rate). Successful adaptation of the client models should lead to higher performance of the overall SV system. In addition, we apply the same scheme to mark the upper performance bound where we use data from the same network type both for enrolment and evaluation. Equivalent to this upper performance bound is the lower bound of the EER. Both performance bounds mark the area where the adaptation performance of a SV system will be found practically. Typically the GSM data shows more variation and worse quality which leads to lower performance (higher EER) compared to a system applying only fixed line data both for training and evaluation of the models.

8.3.2 Adaptation using 'per client' data

As we stated above, there are two sources for the offline data that can be used in the adaptation process. The first variant uses transcoded enrolment data from each client. The transcoding process has already been described in section 7.3. The main idea with this adaptation technique is, that the impact of the GSM speech coding on the client's speech is simulated and this artificially generated data is used to adapt the current client model based on fixed line data.

Figure 8.3 depicts the block diagram of this adaptation type. The client's enrolment data is used in a transcoded version to calculate transformation parameters. The collected parameters for each client are denoted as a *transformation set*. The transformation is applied to the client models separately and the transformed client models are stored in the model database as well[1].

[1]A similar technique would be to store the transformation sets and apply the transformation process on demand.

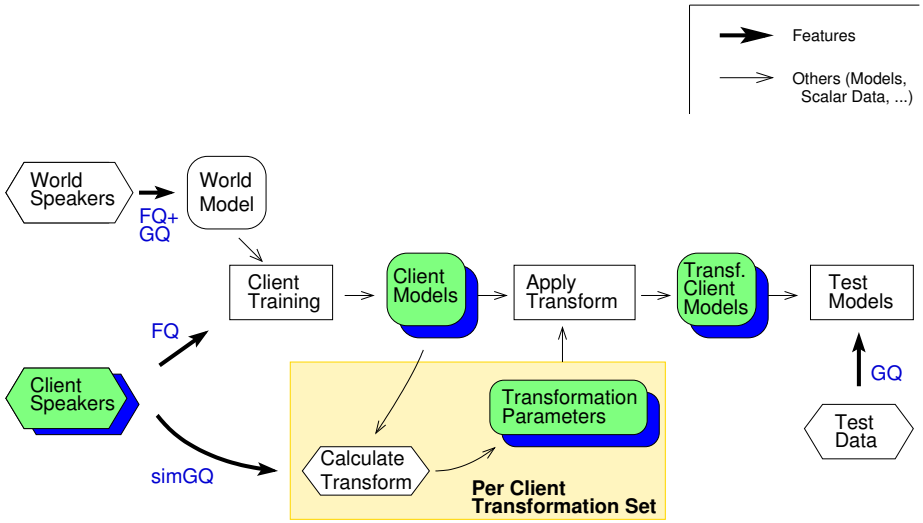

Figure 8.2: Block diagram of the SV system with MLLR using per client transformation sets. Transformation parameters are estimated from simulated GSM recordings (derived from each client's enrolment data).

The adapted models can be used then for the verification process. We will denote this type of transformation set assignment as 'per client'.

We expect that this adaptation system will lead to a better overall performance of the SV system compared to the following adaptation technique as a separate transformation set is assigned to each speaker. However the storage requirements are also higher.

8.3.3 Adaptation using 'common' data

The second source of adaptation data could be an utterance set from a distinctive speaker population. In our case we selected the world speakers for this purpose. Because this data is collected offline during the design phase of a SV system, also recordings of the target network of the adaptation system can be incorporated. With the present data from the world speakers it is either possible to use the 'GQ' data for adaptation or, similar to the 'per client' adaptation system, a transcoded version of the 'FQ' data using the FR GSM speech codec, denoted as 'sim GQ'.

Figure 8.3 shows the necessary components for this adaptation type. We present here the case where the 'sim GQ' speech data is deployed for the adaptation system (the case for using the 'GQ' data set is analogous).

First, the 'FQ' utterances of the world speakers are used to build an auxiliary model, similar to the world model. Next, the transcoded data is taken to estimate the transformation parameters of the auxiliary model. The resulting transformation set is stored and applied in the adaptation of the client models in the SV system. Here we assume that all client models can be transformed by an universal transformation set that we can estimate from an independent model (the auxiliary model). The fact that we have to generate only a single transformation set gives this transformation assignment type the name 'single' transformation set.

Alternatively, we can also deploy the 'GQ' data of the world speakers which might result in a more realistic transformation set. The computational effort and the storage requirements are equal in both cases. If our assumption of a general transformation set for an adaptation from

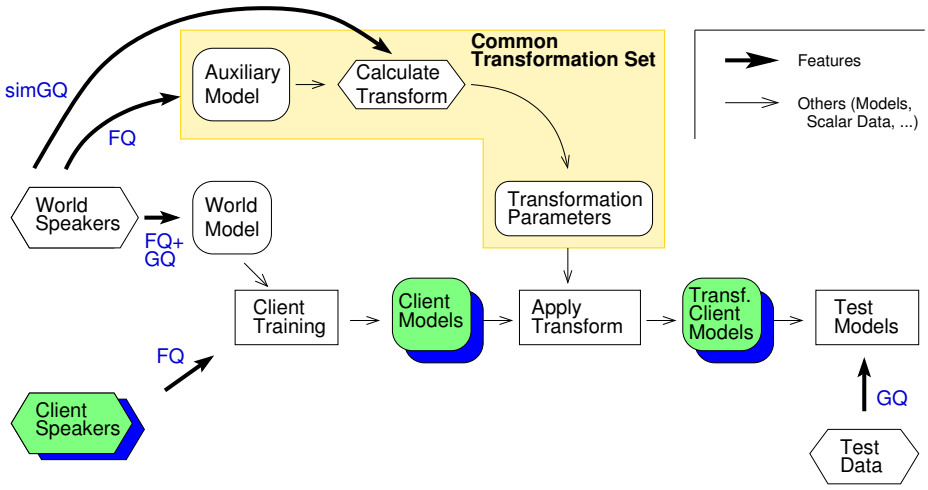

Figure 8.3: Block diagram of the SV system with MLLR using a common transformation set. The transformation parameters are estimated from either simulated GSM recordings (as depicted here) or genuine GSM recordings of the world speakers.

fixed line speech to GSM speech is true, we would expect a higher performance when using 'GQ' data for the transformation estimation compared to the case where we deploy the 'sim GQ' data because more real life effects are contained in the former data subset.

8.3.4 Practically relevant adaptation scenarios

By keeping the basic framework ('FQ' trained base models) fixed, we will use a shorthand notation to describe the various combinations of the type of the adaptation speech material and the type of evaluation speech material: <type of adaptation data>-<type of evaluation data>. To denote the simulated GSM data, we will use the keyword 'sim' and for the real GSM data, we will use the keyword 'real'. The resulting pair will be denoted as *adaptation scenario*. In case of the 'per client' adaptation system presented above, we would write 'sim-real'. This is the only practically relevant adaptation scenario. However, we will use also the scenarios 'sim-sim' and 'real-real' in order to get general performance bounds for the 'per client' adaptation system. Later in section 9.1 we explain this concept in more detail.

The same notation of the adaptation scenario will be used for the 'common' transformation set assignment as well. The two adaptation scenarios that we explained already are sensible for real applications ('sim-real' and 'real-real') while the remaining combination 'sim-sim' serves again for estimating the performance bounds of the adaptation system (see again section 9.1).

8.3.5 Automatic segmentation using speech recognition models

The segmentation of the speech utterances into phonetic units, which is necessary for a phonetic clustering, requires an extra set of phoneme models. Our goal is to show that the adaptation using phonetic classes is generally applicable in an automatic system and supplies additional benefit to model adaptation. Therefore we decided to deploy a set of phoneme models trained on a large corpus of fixed line telephone data and perform an automatic segmentation of the utterances. The models are derived from [88] where they have been trained for a baseline speech

recognition system using telephone data. Apart from small adjustments, the number of phoneme models (43) follows the number of German phonemes according to the SAMPA standard [2].

The phoneme models are trained using data of the SpeechDat (II) corpus [89], where the FDB part containing the fixed line data was selected. More than 150,000 utterances from 3500 speakers incorporate the training corpus. The recordings were made directly from the ISDN speech data channel with A-law encoding, similar to the recordings of the VeriDat corpus. Most of the recordings took place in presumably quiet environments; 71% are recorded in the HOME scenario.

The feature vector contains 12 MFCCs and the logarithmic normalised energy; it is extended by the deviation and the acceleration into a 39-dimensional vector. A separate parametrisation of the VeriDat is necessary for segmentation purposes in order to adapt to the expected features of the models. The segmentation information can be later applied to all types of feature streams. If the parametrization of the SV system applies also silence detection, missing frames in the feature stream must be respected when transferring the segmentation information.

8.4 Related work

Having outlined our proposed system we will continue with presenting related work from literature of adaptation in speaker verification. In this context, we focus on the adaptation of speaker models to data recorded through different handsets or captured from different telephone channels, namely fixed telephone line and cellular data. Most of the presented work deploys HMM-based models, very often text-independent modelling by deploying GMMs. Our selection of publications is restricted to model domain and feature domain compensation. Most of these compensation techniques apply linear transformation functions with the function parameters estimated via a maximum likelihood estimation.

A group around Man-Wai Mak (Hong Kong Polytechnic University et al.) developed several approaches based on *stochastic feature transformation* (SFT), which was introduced by Sankar and Lee [71] for robust speech recognition. We have already given a short outline of SFT in section 8.2. Their SFT variants estimate transformation parameters in the feature domain in order to adapt to different handsets and different coding effects. The progress towards the more recent variants called *constrained stochastic feature transformation* and *blind stochastic feature transformation* allows operation without a dedicated handset selector, that their former SFT approaches required.

Another recent technique which is deployed by other research teams (see e.g. [90, 91, 92]) is the *feature mapping* introduced by Douglas Reynolds, MIT Lincoln Laboratory, in [93]. It is based on a technique called Speaker Model Synthesis which was developed by Teunen et al. [94].

Many of the selected publications use two very popular databases that are frequently selected as standard evaluation corpora for result reporting: HTIMIT and the NIST SRE databases. Both corpora have been presented in more detail in section A.4.1 and A.4.2.

8.4.1 Basic variants of stochastic feature transform (SFT)

The research group mentioned above published several conference papers on the problem of GMM adaptation for robust speaker verification under channel mismatch. Among their work are basic SFT variants [95, 83, 81, 96, 82] and more advanced variants denoted as constrained SFT and

[2] See http://www.phon.ucl.ac.uk/home/sampa/german.htm

blind SFT[87, 86]. In this section, we will present first the basic characteristics of their speaker verification system which is common to their publications. We will proceed with details of their results for the basic SFT variants..

For the basic SFT variants, their concept comprises an automatic handset selector which controls the selection of the precomputed transformation parameter sets. The adaptation technique is based on the stochastic matching concept introduced earlier by Sankar et al. [71] and shares the same basic idea with the classic MLLR adaptation approach [78, 74]. They are thus closely related with each other and some of the linear transformations can be also be realised by MLLR adaptation (see 8.2).

Handset selector

A GMM-based handset selector which is presented in [95, 83] represents a main component in their concept. In speaker verification using adaptation to different environments and channels, the estimation of transformation and adaptation parameters need to be computed without any data from the verification process (see discussion in section 8.2 and [81, section 3]). Hence a reliable selection among different transformation parameter sets must be supplied.

In the cited work the adaptation to channel and handset mismatch is solved by creating sets of transformation parameters for all nine types of possible handsets in the used speech corpus (HTIMIT, see [97]). Each handset type is modeled by a GMM model that results from an unsupervised two-level clustering procedure. In the first level, the EM-algorithm is used to create N clusters from a given set of utterances from various speakers. The number N can be varied and gives the best results when N is about equal to the number of different handset in the corpus. Within a cluster, utterances from handsets with similar characteristics are contained. In the second level, a cluster-specific GMM (Ω_n) is estimated from the utterances associated with a cluster. Linked to each handset GMM is a transformation set which defines the adaptation parameters.

The selection of a particular transformation set $n*$ is done by a maximum likelihood selection based on an observation sequence $\mathbf{O}$:

$$n* = \arg\max_n \sum_{t=1}^{T} \log p(\mathbf{O}|\Omega_n) \tag{8.18}$$

where $p(\mathbf{O}|\Omega_n)$ is the likelihood of the n-th cluster given the observation $\mathbf{O}$.

Corpora characteristics

The investigations of the basic SFT variants uses HTIMIT [97] and up to six versions of a transcoded HTIMIT corpus for training and evaluation purposes [81]. We have described the original HTIMIT corpus already in section A.4.1.

The transcoded versions of HTIMIT are derived from coding and decoding the HTIMIT corpus with up to six different codecs: G.711 (64 kb/s), GSM (13 kb/s), G.729 (8 kb/s), G.723.1 (6.3 kb/s) and LPC (2.4 kb/s). Thus, a channel condition is given by two different domains: the type of handset used during re-recording and the type of speech codec for the recoding (none or any of the six codecs mentioned above).

In the following we will distinguish the *re-recording* of HTIMIT through different handsets from the *transcoding* of this data by different speech codecs.

Feature vectors from these transcoded corpora contain 12-th order mel-frequency cepstra (MFCC) including velocity coefficients at a frame rate of 14ms. In an earlier publication [83] LP-derived cepstral features (LPCC) were used, but did not achieve the same performance than the MFCC features.

Model construction

The genuine speaker sets comprise 100 speakers (50 male, 50 female) while the impostor set uses 25 male and 25 female speakers. Typically, enrolment takes place using the clean data from the Sennheiser microphone (labeled *senh*). Each client GMM uses 32 mixtures and is trained on HTIMIT's SA and SX sentence sets (seven sentences in total). A background GMM using 64 mixtures was trained on all SA and SX sentences of all speakers.

Verification process

Verification takes place by using the commonly used score normalisation with a background model (world normalisation). The normalised score $S(\mathbf{O})$ is computed as

$$S(\mathbf{O}) = \log p(\mathbf{O}|\lambda_c) - \log p(\mathbf{O}|\lambda_w) \tag{8.19}$$

where λ_c is the model of the claimed speaker and λ_w is the background model. For each speaker the threshold was individually adapted to compute the EER. Results are reported as the mean EER over all client speakers[3].

Comparison of transformation schemes

The first publications of Man-Wai Mak et al. [81, 83] uses stochastic matching as transformation scheme which was introduced by Sankar [71]. We presented this technique already briefly in section 8.2. Here, a deterministic bias vector ($\sigma_b = 0$) is applied in the feature domain:

$$\hat{\mathbf{o}}(t) = \mathbf{o}(t) + \boldsymbol{b} \tag{8.20}$$

For parameter estimation the clean data of 10 speakers is used to train a 2-mixtures GMM ($M = 2$ in formula 8.15). The clean model serves then in the estimation process of the vector $\boldsymbol{b}$ where the transcoded data of the same 10 speakers is applied.

Previous work in [83] showed that a general second-order transformation $\hat{\mathbf{o}}(t) = \boldsymbol{c}(\mathbf{o}(t))^2 + \boldsymbol{a}\mathbf{o}(t) + \boldsymbol{b}$ can be replaced by an equally performing zero-th order transformation $\hat{\mathbf{o}}(t) = \mathbf{o}(t) + \boldsymbol{b}$. Even the use of piece-wise defined transformation functions of zero-th order up to second order showed no performance gain.

For each handset cluster n an associated bias vector $\boldsymbol{b}_n$ is calculated and applied to all observations of the utterance currently under test.

Results

To summarise the results of [82, 81, 72] briefly, we will concentrate on their results for both the original HTIMIT corpus and the GSM transcoded HTIMIT corpus. Please note that the speaker

[3] In later work which is presented in the next sections, the EER is computed from the DET plot according to the NIST SRE guidelines.

models are build from the clean *senh* data and that the evaluation is performed with data from different handset types while the transcoding type is always the same within a comparison.

Training and evaluation was performed independently both on the uncoded corpus and its transcoded versions. No cross-comparison between different codings takes place. Adaptation is only performed with regard to the different handset types.

The results for the zero-th order SFT compared with the baseline system (no adaptation) and a CMS adapted system are given in table 8.1.

	Equal Error Rate (%)		
Codec	**Baseline**	**CMS**	**SFT**
Uncoded	9	5	3
GSM	11	6	5

Table 8.1: Selected results taken from [81, figure 1] from the comparison of the overall EER of a baseline system, CMS approach and zero-th order stochastic feature transformation (SFT).

Without any adaptation, the baseline system shows the highest error rates in this comparison when performing a test under mismatched conditions for both the uncoded and the GSM coded corpus. Applying a simple CMS algorithm results in a large improvement. However, as expected, CMS inherently degrades performance also under matching conditions (see [81, table 1,2 and 3]). The use of a single bias for compensation (denoted as 0th-order stochastic transformation) yields a further reduction of the EER. The handset selector accuracy for GSM transcoded data is reported as 96.91% when ten handset clusters are deployed.

Later work extends the presented framework. In [82] the influence of the number of clusters for the handset detection is investigated. It turns out that the cluster based selector can achieve the same performance (measured as average EER over all speakers) as a selector working with an explicitly known handset. The comparison of several transcoded data is reduced to the usage of the GSM transcoded corpus. As found in [81], the error reduction of CMS can be outperformed by a simple zero-th order stochastic feature transformation (SFT). In addition, the property of CMS to increase error rates in matching conditions does not show up.

A more diverse comparison of different adaptation techniques can be found in [72]. Among them SFT with the same transformation function compared to the previous publications in [82, 83, 81] is investigated. In addition, two model transformation or rather adaptation techniques are included into the comparison: MLLR and PDBNN (Probabilistic Decision Based Neural Network, see [98]).

For the MLLR, a first order adaptation of the means is chosen ($\hat{\boldsymbol{\mu}} = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b}$). The comparison of the PDBNN, SFT and MLLR adaptation, together with simpler techniques including CMS, Tnorm [36] and Hnorm [35] reveals most of the already known results. SFT and MLLR show the largest error reduction (average EER of 6.98% and 6.67% resp.) compared to the CMS adaptation of 11.03%.

PDBNNs models perform in between the former two model types (8.44%). Their lower performance compared to MLLR is stated as a possible result of its handset-dependent model that are based on single speaker data. In contrast MLLR and SFT pools the data for estimating the transformation parameters from a large number of speaker (20 to 100 speakers in case of [72]).

In addition, it is found that due to the larger number of parameters for MLLR (matrix $\boldsymbol{A}$ and vector $\boldsymbol{b}$) the convergence over the number of speakers used for parameter estimation is slower compared to SFT.

We highlight again the missing cross-coding evaluation, since in the cited work only handset effects are investigated in direct comparison.

8.4.2 Constrained SFT (CSFT)

In more recent work [86] the constraint of using precomputed transformation sets chosen by a handset selector is tried to be released. Instead of estimating the parameters sets from a large number of adaptation data, the transformation is calculated from the test utterance itself. The remaining parts of the setup are inherited from the earlier setup, that we described already in section 8.4.1.

Regarding the discussion from section 8.2, the problem of adapting a client model towards impostor speech must be taken into account. The unsupervised adaptation technique CSFT uses an implicit constraint while calculating the parameters. The transformation function $\hat{\mathbf{o}}(t) = \boldsymbol{A}\mathbf{o}(t) + \boldsymbol{b}$ is estimated based on the statistical difference between the test utterance and a composite GMM which includes both the mixtures of the client model and the background model (λ_c).

The test utterance itself is modelled by a GMM λ_t with the same number of mixtures as the combined model λ_c. For fast and robust estimation of λ_t, it is generated from a background model that is adapted via MAP adaptation using the test utterance. Finally, the SFT parameters are calculated by using the test utterance and both models λ_t and λ_c..

The main idea of this technique is that the test utterance is mapped into a region between the mixtures of the speaker and the background models. Thus, the transformed data is neither biased towards the speaker model nor the background model. The results are computed on the 2001 NIST speaker recognition evaluation data which contains data sets from cellular phone speech extracted from the SwitchBoard-II phase 4 corpus. Compared to the previous publications, it is the first evaluation based on real life cellular phone data.

As an additional new feature compared with the former publications, the EER is computed from the DET plots using the NIST cost parameters (A.4.2).

Compared with a simple CMS adaptation which gives an EER of 12.02%, the best type of constrained SFT adaptation reaches an EER of 10.00%.

However, no baseline performance (without any adaptation technique) is given. Also, no performance figures are given for non-constrained SFT adaptation as used e.g. in [81, 82, 72, 83]. The Znorm normalisation introduced in [35] reaches a slightly worse performance with an EER of 10.39%.

8.4.3 Blind SFT (BSFT)

In [87] a further extension of the constrained stochastic feature transformation, labeled *blind stochastic feature transformation* (BSFT) is presented. Compared to [86] the calculation of the model λ_t for the test data is discarded. Instead, the BSFT parameters are directly computed from the distorted features $\hat{\mathbf{o}}(t)$ and the composite model λ_c. For evaluation, gender-dependent background models and thus gender-dependent BSFT parameters are deployed.

Again, the comparison of the EER performance based on the NIST SRE 2001 data set is done between simple CMS adaptation (EER = 11.44%), Znorm (EER = 10.61%) and BSFT (best variant of BSFT giving EER = 9.26%).

As already mentioned in section A.4.2, the NIST SRE 2001 data does not include cross-channel performance tests in its evaluation plan. Cellular phone data is used both for training and evaluation, but no mismatch test is done .e.g. evaluating fixed telephone line trained models with cellular data. However, it is a major step towards judging the performance of real-life SV

systems since in the previous publications deploying HTIMIT, only artificially transcoded GSM data was used.

8.4.4 Feature mapping (FM)

Douglas Reynolds developed an adaptation technique called *feature mapping* in [93] that has already been deployed in some contributions to the NIST evaluation program (see e.g. [90, 91, 92]). Feature mapping extends the idea of *speaker model synthesis* (SMS) which was introduced by Teunen et al. in [94] and has its roots in the work on stochastic matching [71].

We will first outline shortly the key features of SMS, the focus on the feature mapping itself and finally report results of its performance in environment adaptation.

Speaker model synthesis (SMS)

The unique feature of SMS is that verification always takes place with a channel-adapted speaker model and a channel-adapted background model when computing the log-likelihood score. The channel-adapted background models are computed offline from data of known transmission channels by adapting a channel- and gender-independent background model (root model) via maximum a posteriori (MAP) adaptation. The transformations which map the root model to a channel-adapted background model are stored for later use. Since all background models are derived from the same root, a one-to-one relation between individual Gaussian mixtures can be stored.

During enrolment the channel used by the client is detected and the speaker-dependent model is derived from the corresponding channel-dependent background model via MAP. In the verification phase, the channel is again detected and the corresponding channel-dependent speaker model is retrieved for score computation. If the detected channel is not available among the speaker's models, it is synthesized from the background model with the matching channel type by applying the stored transformations from the offline training.

The transformations of the Gaussian mixtures are expressed by the mean shifts, variance scales and weight scales.

Key features of feature mapping

While SMS operates in the model domain, feature mapping transfers the mapping idea into the feature domain. Reynolds suggests a common channel-independent feature space (described by the channel-independent root model) to which features from different channels are mapped. SMS and feature mapping are related as these mappings are learnt by examining the modifications of models that where adapted via MAP. As claimed by Reynolds, mapping in the feature domain can be deployed more generally since it is not dependent on a particular model structure. In addition training data from different channel types can be incorporated into the speaker's model.

Again, as in the SMS approach, a channel-independent root model is trained and channel-dependent background models are derived via MAP using channel-dependent data. The differences between the later models and the root model indicate how the feature mapping must be computed. In [93] the mean shifts and the variance scales determine the feature mapping function. Assuming a diagonal covariance GMM, the parameters of a channel-independent mixture

(mean and standard deviation denoted as μ_{CI} and σ_{CI} resp.) and the corresponding parameters of the Gaussian mixture from a channel-dependent model (μ_{CD1} resp. σ_{CD1}) define the transformation function in the feature space:

$$\mathbf{y} = (\mathbf{x} - \mu_{\text{CD1}})\frac{\sigma_{\text{CI}}}{\sigma_{\text{CD1}}} + \mu_{\text{CI}} \tag{8.21}$$

where $\mathbf{x}$ denotes the features of channel CD1 and $\mathbf{y}$ denotes the features in the channel-independent domain.

During enrolment, the most likely channel-dependent background model determines the transformation set used on the enrolment data. The channel type can be kept fixed for the whole utterance or can be selected over a short-term window in order to adapt to varying conditions. The enrolment data is mapped into the channel-independent feature space and then used for model training. Data from different channels can by easily incorporated in the speaker model.

In the verification phase, the same technique is applied: the channel-dependent background models are used to select the channel type, the features are mapped by the selected transformation and the system then uses the root model as background model and the channel-independent model with the remapped features for computing the likelihood ratio score.

Results

Douglas compares the feature mapping approach with SMS based on data from the NIST speaker recognition evaluations of the year 2000 (fixed telephone line data) and 2001 (cellular data). As we have shown before, these corpora include data from different handset types (carbon-button and electret microphones). The speaker verification system uses GMMs with 2048 mixtures as background models. The channel-dependent models are derived from the root model via MAP adaptation. Performance is reported by using DET plots generated from pooling all scores from male and female trials. No cross-sex attempts are made.

The fixed telephone line data reveals that feature mapping and speaker model synthesis perform nearly similar and both provide a significant performance improvement compared to the baseline system without adaptation (from around EER = 13% to around 10.5%). For the cellular data, the channel independent root model is adapted via MAP prior to building the channel-dependent models. The performance gain here results in an improvement from an EER = 11% to EER = 9.1%.

Again, as we have stated in section 8.4.3, no real cross comparison between the fixed telephone line and the cellular data is reported. The performance gains on more subtle variation in the channel (handset type) will not allow any prediction on the performance on more varying channel types that are given by a mixture of fixed telephone and cellular recordings.

8.5 Model adaptation using MLLR

In the following we will present in more detail the application of MLLR for adapting Gaussian parameters of HMMs. First we will introduce the original definition of the mean transformation by a linear transformation function and its extension with a variance transformation matrix. We will then continue with the presentation of our parameter adaptation which uses bias vectors for mean translation (denoted as 'Bias') and extend it with a variance scaling (denoted as 'Bias + VarScale'). Suggested by our investigations of the impact of GSM coding on cepstral feature

vectors (see section 7.5), independent translation vectors for the cepstral coefficients from different phonetic groups might be used for adapting GSM speech towards non-matching speaker models. Especially the low number of parameters of a bias-only transformation compared with the full linear transformation could give a robust but still effective adaptation scheme.

Finally we will compare the different MLLR types according to their computational complexities and costs.

8.5.1 Application of MLLR for HMMs

The well established MLLR adaptation scheme is based on linear transformations in the model space; its parameters are obtained via maximum likelihood estimation. Originally it was intended for speaker adaptation of HMM-based speech recognizers [74, 75]. In addition, MLLR is successfully applied in speech recognition for environment compensation [71, 99]. In the context of speech recognition model-space linear transformations have been proven to be more appropriate than feature-space transformations. In the MLLR framework there are two main forms of transformation. First, in the unconstrained case, transforms of the means and variances of the models are unrelated with each other. Alternatively, in the constrained case (see e.g. [75]), the mean transformation and the variance transformation are closely related which renders this case very similar to feature-space transformation. In the sequel we will focus on the unconstrained case because it permits a larger set of possible transformation and offers thus more flexibility.

The parameters of the linear transformations are found using the well known EM approach [23]. The parameters of the transform are determined by maximizing the auxiliary function $Q(\lambda, \hat{\lambda})$:

$$Q(\lambda, \hat{\lambda}) = \\ constant - \frac{1}{2} \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t) \left[K + \log(|\hat{\boldsymbol{\Sigma}}_m|) + h(\mathbf{o}(t), m) \right] \tag{8.22}$$

with

$$h(\mathbf{o}(t), m) = (\mathbf{o}(t) - \hat{\boldsymbol{\mu}}_m)^T \, \hat{\boldsymbol{\Sigma}}_m{}^{-1} \, (\mathbf{o}(t) - \hat{\boldsymbol{\mu}}_m) \, . \tag{8.23}$$

The symbols $\hat{\boldsymbol{\mu}}_m$ and $\hat{\boldsymbol{\Sigma}}_m$ denote the transformed mean and variance for the Gaussian component m. There are in total M Gaussian components assigned to the particular transform. They need not necessarily belong to the same state in the HMM. This tying of Gaussians enables robust training of the transformations (see a more detailed description in section 8.6). The set of Gaussians will be later referred to as a *regression class*. The posterior probability, $\gamma_m(t)$ is

$$\gamma_m(t) = p(q_m(t) \,|\, \boldsymbol{O}, \lambda) \tag{8.24}$$

and gives the probability of the Gaussian component m at time t on the given adaptation data $\boldsymbol{O} = \{\mathbf{o}(1)), \ldots, \mathbf{o}(T)\}$. Finally, the constant in 8.22 depends only on the transition probabilities and $K = n \log(2\pi)$ is the normalisation constant for each Gaussian component.

Linear transforms in the unconstrained case

The unconstrained case of the linear model-space transforms allows unrestricted transformations of the mean vector and the covariance matrix. A general linear transform of the mean vector $\boldsymbol{\mu}$ resulting in a new mean vector $\hat{\boldsymbol{\mu}}$ is given by

$$\hat{\boldsymbol{\mu}} = \mathbf{A}\boldsymbol{\mu} + \mathbf{b} = \mathbf{W}\boldsymbol{\xi} \tag{8.25}$$

where $\boldsymbol{\xi}$ is the extended mean vector $\begin{bmatrix} 1 \\ \boldsymbol{\mu} \end{bmatrix}$ and $\boldsymbol{W}$ is the extended transform $[\boldsymbol{b}\ \boldsymbol{A}]$.

The variance transform may be modelled in two forms using either a Choleski decomposition of the original covariance matrix $\boldsymbol{\Sigma}$ or a quadratic form (see [100]). We choose in the following the quadratic form

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T \tag{8.26}$$

where $\boldsymbol{H}$ is the transformation matrix to be obtained.

In the sequel we will briefly describe the formulae for estimating the mean transformation parameters, then we present the estimation of the variance transformation matrix. Our notation follows the one used in [74].

Estimation of the mean transformation parameters

We assume diagonal covariance matrices in the following. The standard auxiliary function given in 8.22 is formulated for the mean transformation 8.25, then maximized with respect to the transformed mean [74]. Finally it is obtained:

$$\sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_m(t){\boldsymbol{\Sigma}_m}^{-1}\mathbf{o}(t){\boldsymbol{\xi}_m}^T = \sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_m(t){\boldsymbol{\Sigma}_m}^{-1}\boldsymbol{W}^{(m)}\boldsymbol{\xi}_m{\boldsymbol{\xi}_m}^T. \tag{8.27}$$

where $\boldsymbol{\xi}_m$ is again the extended mean vector $\begin{bmatrix} 1 \\ \boldsymbol{\mu}_m \end{bmatrix}$ To solve for $\boldsymbol{W}^{(m)}$, we rewrite the left hand side of 8.27 as a matrix

$$\boldsymbol{K} = \sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_m(t){\boldsymbol{\Sigma}_m}^{-1}\mathbf{o}(t){\boldsymbol{\xi}_m}^T \tag{8.28}$$

which is independent of the transformation matrix $\boldsymbol{W}^{(m)}$. For decomposing the right hand side of 8.27, we define the matrix $\boldsymbol{G}^{(i)}$

$$\boldsymbol{G}^{(i)} = \sum_{m=1}^{M}\frac{1}{\sigma_i^{(m)^2}}\boldsymbol{\xi}_m{\boldsymbol{\xi}_m}^T\sum_{t=1}^{T}\gamma_m(t) \tag{8.29}$$

where $\sigma_i^{(m)^2}$ denotes the i-th diagonal entry of the covariance matrix. In addition we rewrite the i-th row vector of $\boldsymbol{K}$ as

$$\boldsymbol{k}_i^T = \sum_{m=1}^{M}\sum_{t=1}^{T}\gamma_m(t)\frac{1}{\sigma_i^{(m)^2}}o_i(t){\boldsymbol{\xi}_m}^T. \tag{8.30}$$

The term $o_i(t)$ indicates the i-th component of the observation $\mathbf{o}(t)$. Equation 8.27 can now be decomposed row-wise:

$$\boldsymbol{k}_i^T = \boldsymbol{G}^{(i)}\boldsymbol{w}_i^T. \tag{8.31}$$

Hence we can calculate the transformation matrix row-wise using

$$\boldsymbol{w}_i^T = {\boldsymbol{G}^{(i)}}^{-1}\boldsymbol{k}_i^T. \tag{8.32}$$

The calculation of each row of $\boldsymbol{W}$ requires the inverse of a $(n+1)\times(n+1)$ matrix which is a $\mathcal{O}(n^3)$ operation. Hence, the total computation needs $\mathcal{O}(n^4)$ operations. This may be contrasted with the cost of the full covariance case which takes $\mathcal{O}(n^6)$ operations.

We will refer to this MLLR type given by equation 8.25 later on as 'Full'.

Estimation of the variance transformation matrix

The optimisation of the variance transform matrix $\boldsymbol{H}$ from equation 8.26 is performed by computing the inverse matrix $\boldsymbol{A} = \boldsymbol{H}^{-1}$ such that

$$\hat{\boldsymbol{\Sigma}}^{-1} = \boldsymbol{A}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{A} \tag{8.33}$$

When the variance transformation is used solely, it can be estimated directly from the observations centralised to the current mean: $\hat{\mathbf{o}}(t) = \mathbf{o}(t) - \boldsymbol{\mu}$. However when the means are transformed in addition, the optimisation of the auxiliary function is done in two stages [99]. First, the mean transformation is estimated given the current variances. Second, the variance transform is estimated from the observations corrected by the already estimated mean: $\hat{\mathbf{o}}(t) = \mathbf{o}(t) - \hat{\boldsymbol{\mu}}$. This process can be repeated iteratively with a monotonically increasing likelihood guaranteed.

The objective is to maximize the equation 8.22 which gives the following expression:

$$\begin{aligned} &Q(\lambda, \hat{\lambda}) = \\ &\quad \text{constant} - \frac{1}{2} \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t) \left[K + \log(|\hat{\boldsymbol{\Sigma}}_m|) - \log(|\boldsymbol{A}|^2) + (\boldsymbol{A}\hat{\mathbf{o}}(t))^T \, \hat{\boldsymbol{\Sigma}}_m{}^{-1} \, (\boldsymbol{A}\hat{\mathbf{o}}(t)) \right] \end{aligned} \tag{8.34}$$

If we assume again diagonal covariance matrices we obtain the solution for the i-th row of the matrix $\boldsymbol{A}$ [100]

$$\boldsymbol{a}_i = \boldsymbol{c}_i \boldsymbol{G}^{(i)^{-1}} \sqrt{\frac{\sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t)}{\boldsymbol{c}_i \boldsymbol{G}^{(i)^{-1}} \boldsymbol{c}_i}^T} \tag{8.35}$$

where

$$\boldsymbol{G}^{(i)} = \sum_{m=1}^{M} \frac{1}{\sigma_i^{(m)^2}} \sum_{t=1}^{T} \gamma_m(t) \, (\hat{\mathbf{o}}(t)) \, (\hat{\mathbf{o}}(t))^T \tag{8.36}$$

and $\boldsymbol{c}_i$ is the i-th row of the cofactors of $\boldsymbol{A}$ and $\boldsymbol{a}_i$ is the i-th row of $\boldsymbol{A}$. The optimisation is thus an iterative one, where each row of $\boldsymbol{A}$ is optimized given the current value of all other rows. The final solution for the variance transform matrix $\boldsymbol{H} = \boldsymbol{A}^{-1}$ is obtained by inversion.

In order to restrict the transformed covariance to a diagonal form, there are two possibilities. First, the non-diagonal elements can be forced to zero. Second, the variance estimation procedure can be modified to directly estimate a diagonal variance transformation matrix $\boldsymbol{A}$ which will then simply scale the diagonal covariance matrix. The latter procedure is used in the current HTK version (3.3 alpha) and is explained in the following.

According to 8.33, a single component of the diagonal covariance matrix will be transformed by

$$\hat{\boldsymbol{\sigma}}_i^2 = \frac{1}{a_{ii}^2} \boldsymbol{\sigma}_i^2 \tag{8.37}$$

Since the cofactors of the matrix $\boldsymbol{A}$ are also diagonal, the term $\boldsymbol{c}_i \boldsymbol{G}^{(i)^{-1}}$ selects the i-th row of $\boldsymbol{G}^{(i)^{-1}}$. In addition it is assumed that the components of the observations are not correlated with each other. This renders the matrix $\boldsymbol{G}^{(i)}$ to the diagonal form. Finally, we receive from equation 8.35

$$a_{ii}^2 = \left(c_{ii} g_{ii}^{-1}\right)^2 \frac{\beta}{c_{ii}^2 g_{ii}^{-1}} \tag{8.38}$$

$$= \frac{\beta}{g_{ii}} \tag{8.39}$$

with $\beta = \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t)$. In contrast to the iterative procedure for estimating the variance transformation matrix from formula 8.35, the simplified version allows to estimate the variance scale in a single step.

Although it is possible to use the variance transformation on its own, we always deploy it in our experiments together with a mean transform. We will refer to the MLLR type using a variance scaling combined with a the full mean transformation matrix as 'Full + VarScale'.

8.5.2 Mean adaptation using a bias vector

The mean adaptation using a full transformation matrix and a bias vector, which we presented earlier, has a computational complexity of $\mathcal{O}(n^4)$. In total there are $n^2 + n$ scalar parameters to be estimated. Using only a bias vector reduces the number of scalar parameters to n but also restricts the flexibility of the possible transformations to simple translation. As we mentioned earlier translations of the feature means e.g. the CMS technique are in spite of their simplicity quite effective. This fact lead us to implement a bias-only MLLR procedure for the adaptation code of HTK.

Estimation of the bias vector

There are two possible ways to estimate a bias vector for the mean transformation. The first possibility is based on a constrained first-order transformation from formula 8.32. The estimation can be restricted for the transformation matrix $\boldsymbol{A}$ being a unity matrix. However for the bias terms we receive an overdetermined equation system whose error could be minimised by using the Gaussian normal equations.

A much simpler way is to reformulate the standard auxiliary function 8.22 using only a bias term. This time we use

$$h(\mathbf{o}(t), m) = (\mathbf{o}(t) - (\boldsymbol{\mu}_m + \boldsymbol{b}))^T \, {\boldsymbol{\Sigma}_m}^{-1} \, (\mathbf{o}(t) - (\boldsymbol{\mu}_m + \boldsymbol{b})) \,. \tag{8.40}$$

To find the maximum of the standard auxiliary function with respect to the bias vector, we write

$$\begin{aligned} \frac{d}{d\boldsymbol{b}} Q(\lambda, \hat{\lambda}) &= -\frac{1}{2} \frac{d}{d\boldsymbol{b}} \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t) \left[K + \log(|\boldsymbol{\Sigma}_m|) + h(\mathbf{o}(t), m) \right] \qquad (8.41) \\ &= 0 \end{aligned}$$

Because $h(\mathbf{o}(t), m)$ is the only term depending on the bias vector, we receive

$$\begin{aligned} \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t) \left[\frac{d}{d\boldsymbol{b}} h(\mathbf{o}(t), m) \right] &= 0 \\ \Leftrightarrow \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t) \, {\boldsymbol{\Sigma}_m}^{-1} \, (\mathbf{o}(t) - (\boldsymbol{\mu}_m + \boldsymbol{b})) &= 0 \\ \Leftrightarrow \sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t) \, {\boldsymbol{\Sigma}_m}^{-1} \, (\mathbf{o}(t) - \boldsymbol{\mu}_m) &= \left[\sum_{m=1}^{M} \sum_{t=1}^{T} \gamma_m(t) {\boldsymbol{\Sigma}_m}^{-1} \right] \boldsymbol{b} \qquad (8.42) \\ \Leftrightarrow \boldsymbol{d} &= \boldsymbol{E}\boldsymbol{b}. \qquad (8.43) \end{aligned}$$

In the last line we wrote both required statistics as a vector $\boldsymbol{d}$ and a diagonal matrix $\boldsymbol{E}$. Hence we can solve for the bias vector:

$$\boldsymbol{b} = \boldsymbol{E}^{-1}\boldsymbol{d}. \tag{8.44}$$

If we assume again the case of a diagonal covariance matrix, 8.44 is decomposed into n independent equations because $\boldsymbol{E}$ takes diagonal form. This type of MLLR transform is later referred to as 'Bias'.

Combination with variance scaling transformation

Here we use again the technique explained in 8.5.1 where the joint maximisation with respect to the mean and the variance transformation parameters is done in two steps. The simplified estimation process for the variance scaling from equation 8.39 is deployed here again. The centralised observations $\hat{\mathbf{o}}(t)$ required in equation 8.36 depend now on the translated mean:

$$\hat{\mathbf{o}}(t) = \mathbf{o}(t) - (\boldsymbol{\mu} + \boldsymbol{b}). \tag{8.45}$$

We will later refer to the combined transformation of the mean with a bias and combined variance scaling as 'Bias + VarScale'.

8.6 MLLR regression classes

As with any parameter estimation problem, the estimation of the transformation parameters in the MLLR framework must be robustly performed given the available adaptation data. The most often used approach to ensure this requirement is to cluster Gaussian mixtures from the HMM models depending on the available data. These clusters are called *regression classes* and each of them is assigned to separate transformation parameter set.

In our previous presentation of the MLLR parameter estimation, we used a simplified notation when we assumed that M Gaussian mixtures share the transformation $\boldsymbol{W}$. If we apply more than one regression class, we denote a particular transform $\boldsymbol{W}_c$ with the index c, each associated with M_c Gaussian mixtures. Typically, each mixture is assigned to exactly one regression class (hard assignment). In [101], an alternative method called *soft assignment* is presented.

The general idea behind this scheme is that the Gaussian mixtures can be divided into distinct groups that are transformed in a similar fashion due to a change by the applied speech codec.

Typically regression classes are determined dynamically based on the amount of adaptation data available by using a regression tree [102]. A binary tree is used to define a hierarchy of base classes (containing typically single Gaussian mixtures) at the leaves and regression classes at the nodes. Alternatively the regression classes can be pre-determined and kept fixed prior to adaptation.

The use of the regression tree allows more flexibility for different amounts of adaptation data for each speaker or each environment. When using a fixed setup of only a few regression classes together with a controlled set of adaptation material there is no need for a regression tree. Our adaptation system deploys thus a pre-determined assignment of Gaussian mixtures to regression classes.

Clustering of Gaussian mixtures is normally done either based on distance criteria in the acoustic space or based on phonetic knowledge. Both schemes are presented in more detail in the next two sections.

8.6.1 Regression classes based on acoustic clustering

The main assumption used with acoustic clustering is that Gaussian mixtures located close with each other in the acoustic space will be changed by environment and channel effects in a similar manner. In contrast to the phonetic clustering this is a data driven approach with no need to provide a phonetic segmentation and phonetic expert knowledge.

Typically the concept of 'closeness' in the acoustic space is measured by the Euclidean distance between the means of two Gaussian mixtures. The HTK tools use a variant of the LBG (Linde, Buzo, Gray) algorithm for the clustering (see [28]). Starting with the root node containing all Gaussian mixtures, the overall mean is perturbed by ±20% of the overall standard variance. The binary splitting is performed until the desired number of clusters is reached.

8.6.2 Regression classes based on phonetic clustering

The assignment of a mixture to a phonetic class can give an alternative clustering scheme which is based on expert knowledge. The phonetic classes can be either very specific (using classes such as front vowels, back vowels, etc.) or rather broad (e.g. vowels, fricatives, etc.).

The main idea behind using a phonetic clustering is, that some phonetic classes might be acoustically very similar but could be affected by different forms and grades of artifacts by a speech coding technique. Especially the two phoneme groups of fricatives and nasals that show the largest sensitivity to GSM speech coding can thus be separated from the remaining phonemes.

The partition into phonetic classes in our experiments is performed in a very broad manner using either six or two different classes. Figure 8.4 shows the two types of partitions. Apart from the pauses, five main phonetic classes (vowels, fricatives, plosives, approximates and nasals) constitute the basic partition. We have already used this clustering in section 7.5 when examining the effects of the GSM codec on basic phonemes categories. Further merging of the six phoneme classes leads to the clustering scheme with a single class for 'nasals' and 'fricatives' (abbreviated NF) and the remaining classes (denoted nonNF).

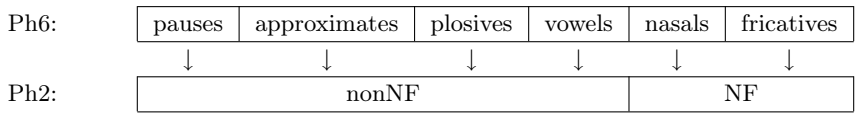

Figure 8.4: Phonetic classes used to build the clustering schemes 'Ph6' and 'Ph2'

The clustering of mixtures can now be performed by assigning each of them to exactly one phonetic class from the basic set Ph6. The second type of clustering using the dichotomous set Ph2 can be gained by further joining of the basic mixture groups.

The assignment is performed differently depending on the type of models used: for the HMM subword models we use an implicit assignment while for the GMM models we calculate an explicit assignment for each mixture. In the next two subsections we present both types of mixture assignment in more detail.

Construction of regression classes for subword HMMs

In this case we deploy the fact that the HMM subword models have been designed to contain one Markov state for each phoneme occurring in the canonical realization of the subword. The

assumption is that after model training each of the states represents the acoustical properties of the phoneme for which it is intended for. Thus the Gaussian mixtures of a state can be assigned to one phonetic class from the basic set Ph6.

We denote with *implicit assignment* the underlying assignment of mixtures to phoneme classes that is defined already by the model. When setting up the model structure, each HMM state in the subword model is intended for a particular phoneme.

In reality not only data from the intended phoneme type but additional phonemes are modelled by a single HMM state. A manual verification of some utterances revealed that basically this automatic segmentation produced reasonable results. However, no quantitative comparison with a reference segmentation from e.g. manually segmented data has been made.

Two facts might explain the rather good performance of this automatic segmentation: the use of the qualitatively best recordings in the database (FixedQuiet) and the short lengths of the subwords units.

Construction of regression classes for GMMs

In the case of the GMM model we can not rely on a predefined mapping of model states to phonetic segments of the utterances. Our proposal is to assign each of the mixtures of the GMM to a phonetic class by taking the mixture occupation probability into account. We will describe the algorithm more detailed in the following:

The training material that was used to estimate the parameters of the GMM is automatically segmented, using the speech recognition models as described in section 8.3.5. The segments from the training data are merged into groups defined by the broad phonetic class definitions 'Ph6' or 'Ph2'. Thus we receive either six or two subsets of the acoustic training data denoted as $\mathcal{C}_k$ with k denoting one of the corresponding six respectively two class labels. For each of the subsets we perform a Baum-Welch reestimation step for the mixture weights. Then each mixture is assigned to the phonetic class for which the corresponding training subset gave the highest reestimated weight.

We simplify the notation of the Baum-Welch estimation step for the case of a GMM using M mixtures:

The reestimated mixture weight $\hat{w}_m$for mixture m is

$$\hat{w}_m = \frac{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma_m^r(t)}{\sum_{r=1}^{R}\sum_{t=1}^{T_r}\gamma^r(t)} = \frac{\mathrm{occ}_m}{\mathrm{occ}} \tag{8.46}$$

which is defined as ratio of a two accumulated occupation probabilities on the given training data. The term occ_m denotes the accumulated occupation probability of the mixture m; the occupation probability $\gamma_m^r(t)$ itself describes the probability of occupying the m'th mixture component at time t for the r'th observation. The accumulated state probability is given by

$$\gamma^r(t) = \sum_{m=1}^{M}\gamma_m^r(t). \tag{8.47}$$

In a single state model this results in $\gamma^r(t) = 1$ because all observations are assigned to the single state. Thus occ contains the number of training observations. For a GMM, the occupation

probability of mixture m is given by (see appendix D)

$$\gamma_m^r(t) = \frac{w_m b_m(\mathbf{o}^r(t))}{b(\mathbf{o}^r(t))} \tag{8.48}$$

with w_m being the original weight of the mixture.

In order to describe the dependence of $\hat{w}_m$ on the training subset we will denote the reestimated mixture weight as $\hat{w}_m(\mathcal{C}_k)$.

For each mixture m in the model:

For each training subset $\mathcal{C}_k$:

Calculate $\hat{w}_m(\mathcal{C}_k)$.

Assign mixture m to the regression class $\mathcal{M}_K$ with phonetic class $K = \arg\max_k \hat{w}_m(\mathcal{C}_k)$.

Table 8.2 shows the regression classes based on the six basic phoneme groups computed for a 32 mixture GMM using the training utterances S2_FixedQuiet_training. The results for two different GMMs are listed: the first one taken from client speaker 0001 (adaptation setup using 'per client' transformation parameters). The second one is the auxiliary GMM used for the 'common' transformation case.

In both cases the phonetic groups are evenly populated with mixtures; none of them is modelled by a single or even no mixture. In addition the assignment of mixtures to phoneme groups seems to be quite consistent when comparing the composition of the mixture classes between the transformation of speaker 0001 and the common transformation.

	Transform speaker 0001		Common transform	
Phonetic group k	No. of mixtures	Mixture class $\mathcal{M}_k$	No. of mixtures	Mixture class $\mathcal{M}_K$
Approximates	5	3, 17, 25, 27, 29	6	2, 3, 17, 25, 27, 29
Fricatives	6	2, 4, 6, 22, 24, 31	5	4, 6, 8, 22, 31
Nasals	3	7, 9, 32	5	5, 7, 9, 15, 32
Pauses	6	8, 10, 12, 14, 16, 18	5	10, 12, 14, 16, 20
Plosives	4	20, 26, 28, 30	5	18, 24, 26, 28, 30
Vowels	8	1, 5, 11, 13, 15, 19, 21, 23	6	1, 11, 13, 19, 21, 23

Table 8.2: Assignment of mixtures to phonetic groups. Results computed for 32 mixture GMM using training utterances 'S2 FQ training'.

The same property can also be depicted from figure 8.5 and 8.6. Here for each mixture class $\mathcal{M}_k$, the reestimated mixture weights are averaged over all contained mixtures. The different training subsets $\mathcal{C}_k$ used to calculate the reestimated mixture weights are plotted on the x-axis. Since the training subsets are disjoint with each other, the average mixture weights sum up to unity within each row (i.e. within a mixture class). The distribution of the average mixture weights shows a distinct peak on the matching training subset. All maximum average weights lie above 0.5. The remaining subsets invoke only small average weights. Only for the phoneme group of plosives, the assignment is less clear; the second largest average weight is 0.19 on the fricatives training subset.

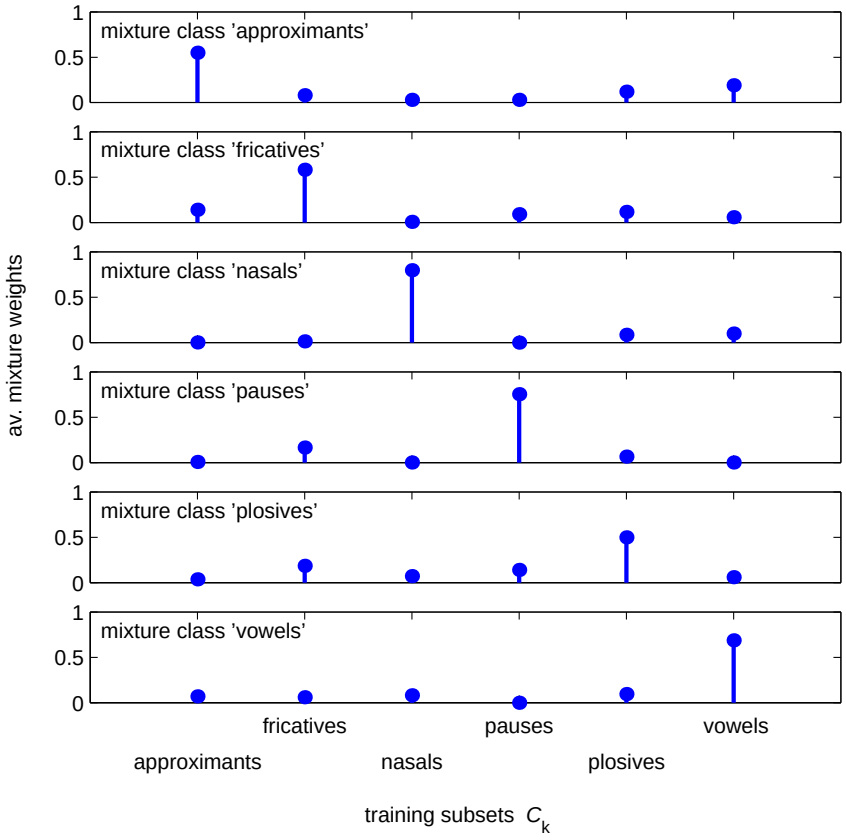

Figure 8.5: Speaker model of speaker 0001: average mixture weights of all mixtures assigned to a particular mixture class (rows) depending on the training subset (plotted on the x-axis). Training utterances taken from speaker 0001, training set 'S2 FQ training'.

The peak in the average mixture distribution becomes less prominent when using the 'common' transformation set. Since it is computed using the auxiliary model and the set of 30 world speakers, more variation due to gender and different speaking styles is introduced. The peak in the distribution becomes less prominent. Again the mixture class of the plosives shows the smallest maximum average weight (0.38) and the broadest distribution over the remaining training subsets. The second largest average weight in this class is found again in the fricatives training subset and its weight (0.19) is below the half of the maximum value.

A practical measure for the prominence of the peak in the distribution of the average mixtures is the entropy. As the mixture weights sum up to unity within a mixture class, they can be regarded as a probability distribution. The entropy of this distribution P is defined as:

$$H(P) = -\sum_{k=1}^{K} p_k \log_2(p_k) \quad \text{with} \quad P = \{p_1, \ldots, p_K\} \tag{8.49}$$

The optimum would be a single peak in the mixture weights within each mixture class; the same distribution would be desirable for the averaged mixture weights. However due to the infinite tails of the Gaussian mixtures this ideal weight distribution can only be reached asymptotically. The entropy in the ideal case will be zero i.e. no information is needed to describe this type of distribution.

The contrary case occurs when no assignment of the mixture to a phonetic group can be made

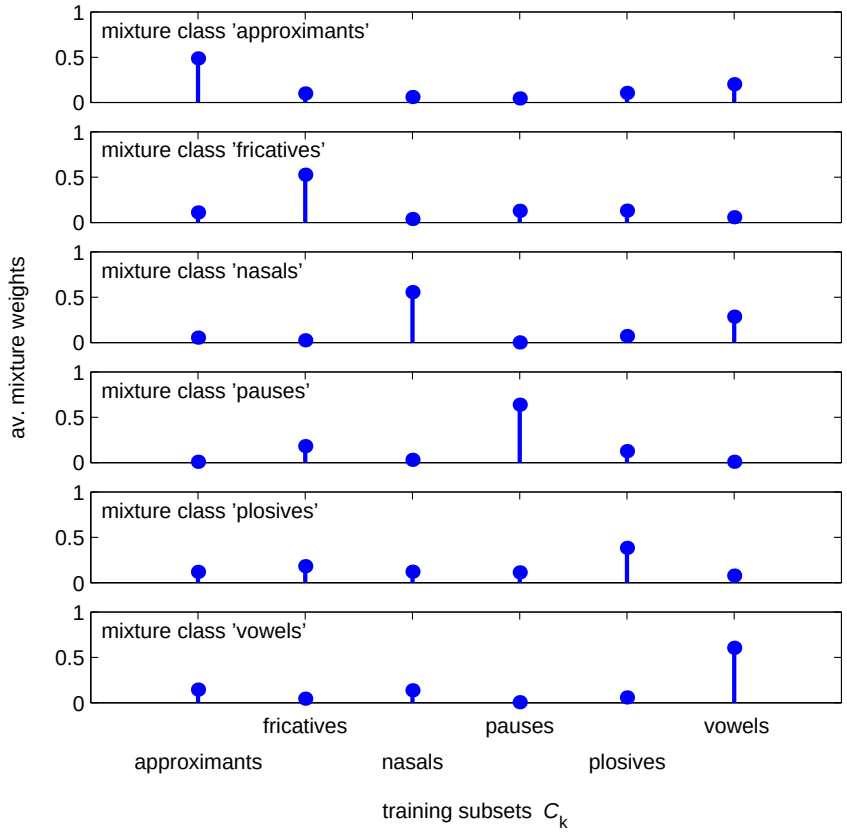

Figure 8.6: Auxiliary model: average mixture weights of all mixtures assigned to a particular mixture class (rows) depending on the training subset (plotted on the x-axis). Training utterances taken from all world speakers, training set 'S2 FQ training'.

which means that the weights on all K training subsets are equal to $1/K$. The entropy then reaches its maximum.

$$H(I_K) = -\sum_{k=1}^{K} \frac{1}{K} \log_2 \left(\frac{1}{K} \right). \tag{8.50}$$

In order to compare entropy values for different numbers of phonetic groups (K), we normalise the entropy on its possible maximum and receive the normalised entropy lying within the range $[0, 1]$.

$$H_n(P) = \frac{H(P)}{H(I_K)} \quad \text{with} \quad I_K = \left\{ p_1 = \frac{1}{K}, \ldots, p_K = \frac{1}{K}. \right\} \tag{8.51}$$

Table 8.3 presents the normalised entropies for the mixture distribution of the six basic phonetic groups 'Ph6', while table 8.4 shows the same results for the dichotomous partitioning 'Ph2'.

Both tables show again the two already mentioned properties of the distributions. First, the speaker dependent assignment leads to more prominent weight distributions (lower entropy) compared to the speaker independent assignment. Second, the assignment of mixtures to the "plosives" class is based on the broadest weight distribution (higher entropy) compared to the assignment for the remaining classes.

Please note that the entropy is highly non-linear. The degradation of the peak prominence in the speaker dependent case (see figure 8.5) going from the class "nasals" ($H_n = 0.39$) to the class

Phonetic group k	$H_n(P)$ for transform speaker 0001	$H_n(P)$ for common transform
Approximates	0.73	0.81
Fricatives	0.70	0.78
Nasals	0.39	0.64
Pauses	0.42	0.59
Plosives	0.79	0.91
Vowels	0.59	0.67

Table 8.3: Normalised entropies $H_n(P)$ computed for the disjoint partitioning of the mixtures into six basic phonetic groups ('Ph6'). Results are given for a speaker dependent assignment (here speaker 0001) and a speaker independent assignment.

"fricatives" ($H_n = 0.70$) seems visually not as strong as a degradation in the upper entropy range when going from the class "fricatives" to the class "plosives" ($H_n = 0.79$).

For the dichotomous phonetic partitioning Ph2 we present the normalised entropies in table 8.4 which are comparable with those from the partitioning Ph6.

Phonetic group k	$H_n(P)$ for	
	transform speaker 0001	common transform
NF	0.83	0.91
nonNF	0.70	0.86

Table 8.4: Normalised entropies $H_n(P)$ computed for the dichotomous partitioning of the mixtures into two basic phonetic groups ('Ph2'). Results are given for a speaker dependent assignment (here speaker 0001) and a speaker independent assignment.

Our results justify the assumption that the mixtures in the GMM can be assigned to broad phonetic groups. The assignment can be done by selecting the maximum reestimated mixture weight when applying specialised training subsets with data from each phonetic group.

Chapter 9

Experiments for GSM channel adaptation

In the previous two chapters we presented the motivation and the proposal for a MLLR based adaptation scheme for a speaker verification system using both fixed line data and GSM data. This chapter describes the experiments for exploring the performance of the proposed adaptation system and presents the results.

The first part contains a detailed presentation of the training and test protocols used for the evaluation and the parameters that are varied in the different experiments.

In the second part, we will outline the sequence of experiments that were conducted.

The next three sections then comprise the experiments in more detail. Two different types of the speaker verification systems are deployed: a text-independent SV system using GMM models and a text-dependent (more precisely: a text-prompted) SV system using subword HMM models. The results for both types are quoted in separate parts and compared with each other. The final and third section presents a more advanced investigation that was done only for the GMM system.

9.1 Experimental setup

Training and test protocols

In section 4.6 we have presented already some training and test protocols that are used for the evaluation of the baseline SV system. The evaluation of the proposed adaptation system poses slightly different requirements to the item selection:

1. Use only the subset containing quiet recordings.
2. If possible, use only single-session enrolment data.
3. Select linguistically matching adaptation data compared to the enrolment data.
4. Evaluation data must neither contain training data nor adaptation data.
5. Restrict evaluation data to either simulated GSM or real GSM data.

The last point is discarded for an additional type of tests. In order to investigate the lower performance bound of the adaptation system we compute the EER of a SV system both trained and evaluated on fixed line data. This gives an additional evaluation scheme without an adaptation data set.

According to the terminology from page 43, we use the 'separate impostors' evaluation protocol.

		Session sets	
Split	**List type**	FQ / simGQ	GQ
		Triplets	
S1	training and adaptation resp.	'S1 FQ training' or 'S1 simGQ training' '01'	'S1 GQ training' '02'
	evaluate	'S1 FQ evaluate' '03', '05', '09', '12', '13', '17'	'S1 GQ evaluate' '06', '08', '11', '15', '18', '20'
S2	training and adaptation resp.	'S2 FQ training' or 'S2 simGQ training' '03','05'	'S2 GQ training' '15', '20'
	evaluate	'S2 FQ evaluate' '01', '09', '12', '13', '17'	'S2 GQ evaluate' '02', '06', '08', '11', '18'
		F1 items	
–	training and adaptation resp.	'F1 FQ training' or 'F1 simGQ training' '01','03', '05', '09'	'F1 GQ training' '02','06', '08'
	evaluate	'F1 FQ evaluate' '12', '13', '17'	'F1 GQ evaluate' '11', '15', '18', '20'

Table 9.1: Session sets for training material, adaptation material and evaluation data. The lists for the triplet experiments (upper part) are split using two variations: split S1 and split S2. For the application of the F1 items (lower part), only one split is used. The session sets for training and adaptation are based on the same session selection (transcoded data is derived from the training material). Only utterances from the 'Quiet' recording sets are used.

Table 9.1 shows the session sets defined for training, adaptation purposes and evaluation. The lists for experiments deploying the triplet items of the VeriDat corpus are described in the upper part, while the lists for the experiments using the fixed utterances (F1 items) are presented in the lower part. As we stated already before, we restricted ourselves to the data from the 'Quiet' subset of the database. In the third column of this table, the session sets (e.g. 'S1 FQ training') are given together with the incorporated sessions. The set identifiers are later used to reference a particular data set.

The triplet sets have been defined in two versions: the split type 'S1' is the initial partition where the sessions were selected in chronological order. During preliminary experiments, we found that the frequency of the number sub-words are highly unbalanced and therefore the sub-word HMM models are not trained equally. We presented this split type here only for completeness. A new split type, denoted 'S2', uses two sessions for training and adaptation respectively which were selected in order to achieve a balanced distribution of the sub-words. The trade-off is, that the selected lists do not respect the chronological order of the sessions anymore. This is an inherent drawback of the arrangement of triplets in the sessions that we already discussed on page 145.

For the F1 items we do not have to consider the linguistic content of individual utterances. Each session contains a single item. Thus data from either four or three chronologically ordered sessions are used for training and adaptation. Preliminary experiments showed that at minimum three utterances should be used for enrolment.

The session assignment for the 'FQ' and for the 'sim GQ' data can be used interchangeable since the simulated GSM data is derived from the fixed line data. To exemplify this, we present the lists required for an adaptation system using 'per client' transformation sets and the adaptation scenario 'sim-real'. In case of the triplet items (using split type 'S2') we

- use 'S2 FQ training' to train each speaker model initially,
- use 'S2 simGQ training' to adapt the model to the GSM recordings,
- evaluate using 'S2 GQ evaluate' in order to receive the performance under real GSM test data.

Varied parameters

Our experiments were conducted for both a GMM-based SV system and a subword HMM-based SV system. In addition, there are in total four degrees of freedom resulting from the properties of individual components in the adaptation system. Table 9.2 summarises them.

Degree of freedom	Possible values
Transformation set assignment	'common' or 'per client'
Adaptation scenario	'sim-sim', 'sim-real' or 'real-real'
MLLR type	'Bias', 'Bias + VarScale', 'Full' or 'Full + VarScale'
Clustering type	'single', 'Ac2', 'Ac6', 'Ph2' or 'Ph6'

Table 9.2: Varied parameters and value ranges used throughout the adaptation experiments.

First, we can vary the type of transformation set assignment: either a common transformation set for all client models is used or each client model is associated with an individual ('per client') transformation set. We have presented both variants in section 8.3.

The second parameter for varying is the type of adaptation data used for estimating the transformation sets and the type of test data used for evaluating the performance of the SV system. Taken together we denote them as adaptation scenario. As we have already presented before in section 8.3, the adaptation scenario must be seen in connection with the transformation set assignment. Again we use the notation '<type of adaptation data> - <type of evaluation data>'; an adaptation scenario 'sim-real' e.g. denotes that the adaptation data is from the simulated GSM data, the 'sim GQ FR' recording set, while the evaluation data comes from the real GSM recordings, the 'GQ' recording set. Because we restricted our experiments in the first part to only the Full Rate GSM speech codec for the transcoded data (data set 'sim G FR' from section 7.3) and only data with quiet background, we can abbreviate the keyword for simulated GSM data to 'sim' and for the real GSM data to 'real'. Later we will also deploy the transcoded data sets with simulated channel errors, denoted as 'sim G FR-C', that we also defined in section 7.3.

Only three combinations of the transformation set assignments with the adaptation scenario are practically relevant (see section 8.3): the 'common' transformation set using either the scenarios 'sim-real' or 'real-real' and the 'per client' transformation set using the scenario 'sim-real'.

In addition we add three more combinations that are not available in practical applications but allow us to estimate the upper performance bounds of the adaptation systems that use the three realistic combinations. We postulate that whenever the type of the adaptation data matches with the type of the evaluation data, the performance of the adaptation systems marks the best achievable results. The realistic adaptation scenarios will show a degraded performance compared with their associated non-realistic counterparts. This fact results from the basic properties of our proposed adaptation system that works only with derived speech data, either with simulated coding effects (simulation of the GSM coding) or generalized adaptation data ('common' transformation set derived from a different speaker population).

The combination of the 'per client' transformation together with the adaptation scenario 'real-real' marks the upper performance bound as we use both individual transformation sets and adaptation data matching with the evaluation data. In contrast, the corresponding realistic adaptation scenario 'real-real' can only deploy pooled data for a common transformation set.

The remaining two combinations come from any type of transformation set assignment together with the adaptation scenario 'sim-sim'. Again there is a close match between adaptation data and evaluation data. However, the evaluation using simulated GSM data marks the upper performance bound of the adaptation system using the simulated GSM data for estimating the transformation sets. Table 9.3 gives an overview of the six combinations used here.

No.	Transformation set assignment	Adaptation scenario	Relevance	Comments
1	common	sim-sim	non-realistic	upper performance bound for 2)
2	common	sim-real	realistic	
3	common	real-real	realistic	
4	per client	sim-sim	non-realistic	upper performance bound for 5)
5	per client	sim-real	realistic	
6	per client	real-real	non-realistic	upper performance bound for 3)

Table 9.3: Combinations between type of transformation set assignment and adaptation scenario; not all pairings are practically relevant however they provide an estimation of the achievable performance (upper performance bound).

The remaining variations in our experiments comprise the MLLR type and the clustering type used for building the regression classes. The different MLLR variants have been presented in sections 8.5.2 and 8.5.1 while the clustering types have been described in sections 8.6.1 and 8.6.2. In some of the following figures we will depict the results of acoustic and phonetic clustering separately and label the complexity of the clustering by the number of regression classes. Using only one regression class is identical to the clustering type denoted as 'single'.

9.2 Overview over conducted experiments

9.2.1 Common properties of the experiments

The main criterion used here for judging the performance and comparing different adaptation techniques is the equal error rate (EER). In many of our following plots for depicting the results, the EER is plotted on the ordinate while the varied adaptation parameter is given on the abscissa.

Two horizontal dashed lines mark two performance bounds for the non-adapted case. The matched test comprises a client model training with the same type of data that is used for evaluating the SV system. If e.g. the evaluation is performed with simulated GSM data, the training is done with the same type as well i.e. the training data matches with the evaluation data.

The mismatched test case shows the performance when no adaptation takes place and the client models are trained using the FQ training data. Here the evaluation is always performed with the genuine GQ evaluation data.

Both performance bounds give the theoretically possible range of the performance of the adaptation system using the specified evaluation protocol. We denote the upper performance bound (i.e. a low EER) for the matched evaluation case while the lower performance bound (a high EER) is marked by the mismatched evaluation case.

In section 9.3 and 9.4 we will focus on the results obtained with the transcoded recording set 'sim GQ FR' where no GSM transmission errors are regarded. Only in section 9.3.3 we will present the results for the more elaborated simulation of the GSM transmission including channel errors. There, the adaptation performance using both transcoded recording sets 'sim GQ FR C-0.08' and 'sim GQ FR C-0.13' is discussed.

Due to the large amount of combinations from the different properties of the adaptation scheme, not all results are reported in this chapter. The quantitative results in term of the EERs can be found in appendix G.

9.2.2 Roadmap for the experiments

Our experimental approach is nearly the same for both the SV system based on the subword-HMMs and the GMMs. These experiments are presented in section 9.3 and 9.4 respectively. First we try to adapt the systems to simulated GSM recordings and evaluate them again with simulated GSM recordings ('sim-sim' scenario). Although this type of adaptation is not intended for real world applications we can prove that the proposed adaptation technique works and which combination of the type of clustering, the number of regression classes and the type of transformation set assignment might be promising for further experiments.

The second, more detailed group of experiments comprise the 'sim-real' scenario. Here, real GSM data is used in the evaluation process. In order to judge the dependency of the different adaptation parameters, we first depict the results by varying all four types of experimental parameters (see table 9.2). Then we select the most promising sub group of parameters and investigate them further by varying the parameters in only a limited range.

We will discuss not only the relative performance differences between different adaptation types but also present their absolute performance compared to the performance bounds that we present in section 9.2.1.

The influence of the simulated GSM data type on the adaptation performance was only investigated for the GMM system (section 9.3.3). Here, we present if and to what extent the incorporation of transmission errors in the simulated adaptation data results in more realistic data and thus in a higher adaptation performance.

Since it turns out that the adaptation sets computed for the 'sim-real' scenario are far less successful than those for the 'sim-sim' scenario, we will present in section 9.5 a more detailed analysis of the differences in the transformation sets for a very basic adaptation technique deploying a single transformation set and the MLLR type 'Bias'.

9.3 Results for GMMs

Our first experiments were conducted with a GMM-based speaker verification system with 32 Gaussian mixtures for both client and world models. We start with the performance in the non-realistic case where simulated GSM data is used for evaluation. After that we will present the results of the practically relevant adaptation scenarios.

9.3.1 Evaluation using simulated GSM data

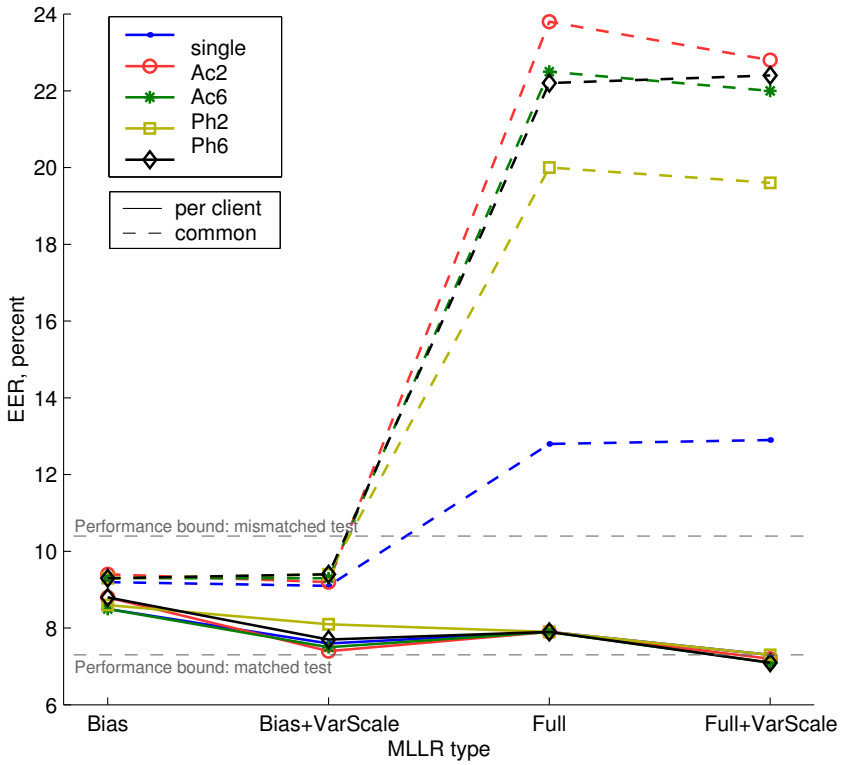

Figure 9.1: Dependence of the EER of the GMM SV system on the adaptation MLLR type (result plot type 1). Adaptation scenario 'sim-sim'. The type of clustering, the number of regression classes and the type of transformation set assignment is varied.

Figure 9.1 depicts the SV performance using the adaptation scenario 'sim-sim'. The dependence on the MLLR type is plotted on the x-axis (going from low complexity MLLR types towards the ones with high complexity) while the transformation assignment type is coded by the line style and the clustering types are encoded by the markers. We will refer to this type of plot as 'result plot type 1'.

We find that the adaptation is generally more successful when using the 'per client' transformation sets compared to the common transformation set. For the former, the best performance values are achieved for the phonetic clustering 'Ph6' together with the MLLR type 'Full + VarScale' and for the acoustic clustering 'Ac2' when using the 'Bias + VarScale' as MLLR type. The performance bound for the matched test case can be reached with these two MLLR types.

The transformation set assignment using a common transformation shows clearly a lower performance than the one using individual transformation sets for each client. However, there is still a performance gain compared to the mismatched test case when using the MLLR types 'Bias' and 'Bias + VarScale'.

A performance even worse than the lower performance bound is reached by adaptation schemes where a full mean transformation matrix is involved. In addition, it seems for these cases that the higher the number of regression classes, the lower the performance falls. This characteristic seems to be unaffected by the clustering type, either acoustic or phonetic.

The reasons for this behaviour remain unclear. We would have expected, that the estimation of a large number of parameters as in the case of the MLLR types with a full transformation matrix might be less robust than the estimation of a bias parameter vector. Especially for the 'per client' set assignment we would expect artifacts due to sparse data. However, our result for the common transformation set, that the adaptation performance degrades with a rising number of regression classes, could be attributed possibly to missing robustness in the parameter estimation due to a decreasing number of observations within a regression class. Since the 'per client' transformation sets are estimated reliably on only the thirtieth part of the data used for the common transformation sets, the large performance differences between the two types of set assignments are probably caused by other effects.

We might conclude that pooling the observations from several speakers in order to calculate a common transformation set introduces too much variability into the source data and thus the resulting adaptation parameters on average do not perform well for all client speakers. In this case, the full transformation matrix seems to be more sensitive to variability than the bias vector.

9.3.2 Evaluation using real GSM data

The next performance overview deploys simulated GSM data for the adaptation and real GSM data for the evaluation (scenario 'sim-real'); it is depicted in figure 9.2.

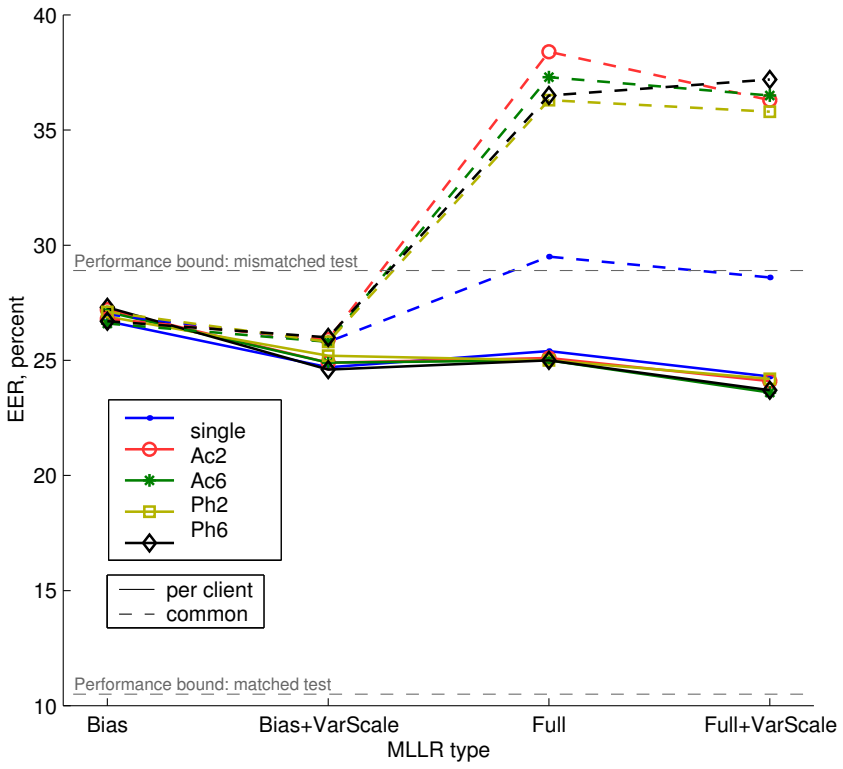

Figure 9.2: Dependence of the EER of the GMM SV system on the adaptation MLLR type (result plot type 1). Adaptation scenario 'sim-real'. The type of clustering, the number of regression classes and the type of transformation set assignment is varied.

Compared to the 'sim-sim' scenario, the performance bounds are higher both for the matched and the mismatched test (10.5% and 28.9% resp.) since real GSM data shows more variability. The resulting models are less selective in the matched case and thus the matched test performance bound rises to higher EERs. Compared to the evaluation with simulated GSM data, the mismatch between fixed line data and the real GSM data is much higher. Hence, higher error rates occur for the mismatched test.

The best performance with an EER of 23.6% can be reached with either acoustic or phonetic clustering using six regression classes and the MLLR type 'Full + VarScale'. We find a large gap between the best adaptation performance and the targeted bound of 10.5% given by the matched test case.

Similar to our findings from the last part of section 9.3.1, the MLLR types involving a full transformation matrix and the common set assignment depict a clear performance deterioration, even below the lower performance bound.

Figure 9.3 depicts the performance of the MLLR types deploying 'VarScale' under the 'sim-real' adaptation scenario when varying the type of clustering and the number of regression classes. This type of plot is later referred to as 'plot type 2'.

We find that applying six regression classes gives the lowest EERs for the two variations using the 'per client' transformation set. For the clustering with six classes, we find a performance gain of 1.0% when going from the 'Bias + VarScale' with 'common' set assignment to the one with 'per client' set assignment and going from there to 'Full + VarScale' MLLR type with the 'per client' set assignment. There seems to be only a weak dependency on the type of the clustering.

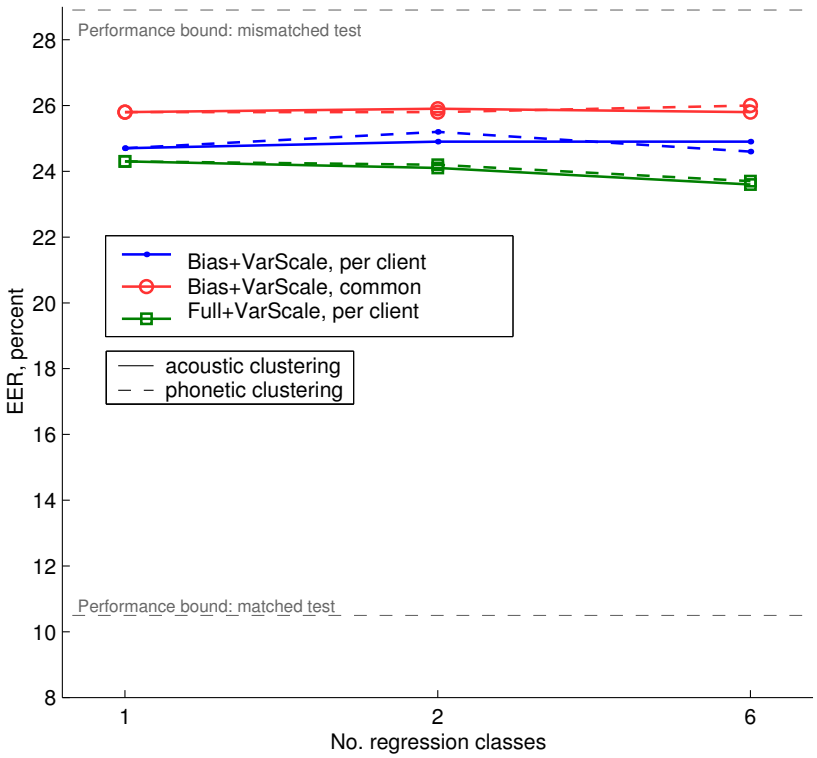

Figure 9.3: Dependence of the EER of the GMM SV system on the number of regression classes (result plot type 2). Adaptation scenario 'sim-real'. The type of clustering is varied for the MLLR types 'Bias + VarScale' and 'Full + VarScale'.

Finally we present a type 2 plot of the MLLR type 'Bias + VarScale' with varying adaptation scenarios in figure 9.4. The three realistic adaptation scenarios show virtually no dependency

on the number of regression classes or the type of phonetic clustering. Still, we can see that the MLLR type 'Bias + VarScale' together with a 'per client' set assignment shows the best performance while the remaining combinations remain in a comparable EER range. The difference between the best and the worst performing adaptation system is around 1.0%.

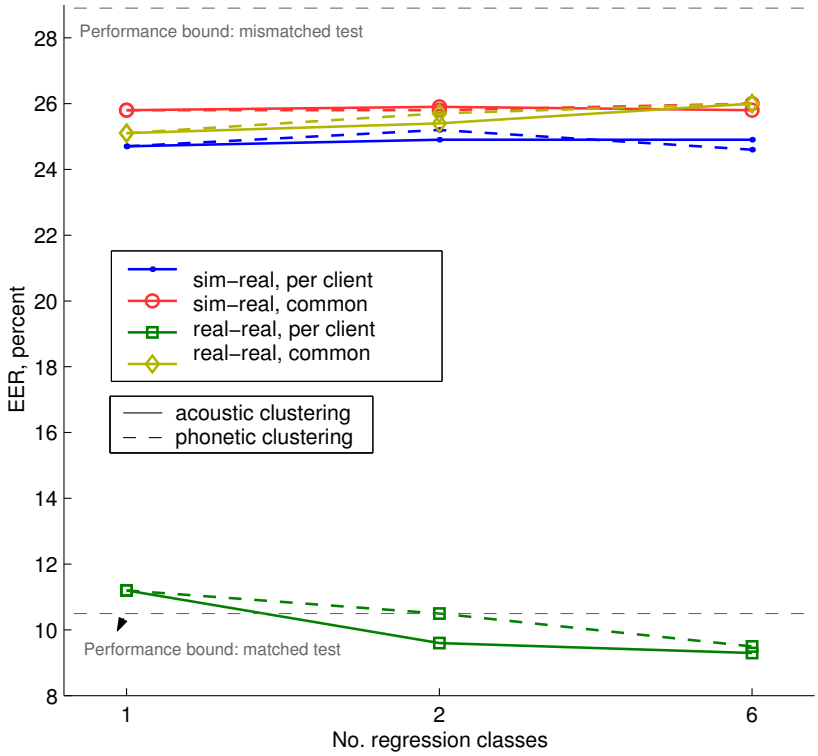

Figure 9.4: Dependence of the EER of the GMM SV system on the number of regression classes (result plot type 2). The type of clustering is fixed ('Bias + VarScale') while the type of adaptation scenario varies.

Despite the fact that we can achieve a small performance improvement of 4% EER compared to the mismatched test case, there remains a performance gap between the best performing adaptation scenario and the lower performance bound of around 14% (absolute difference between EERs).

In addition, figure 9.4 includes also the performance for a non-realistic adaptation scenario where 'per client' transformation sets are calculated from real GSM data (scenario 'real-real'). Here we find that the adaptation to real GSM speech can be successfully accomplished. The fact that the EER values for the 'real-real' scenario with 'per client' assignment fall under the performance bound given by the matched test can be attributed to the selection scheme of the recordings. In the other experiments the adaptation data ('sim GQ FR' set) is based on genuine FixedQuiet recordings that were also used for client modelling. Thus their additional contribution is only the difference between the two recording types. In contrast, the real GSM data set provides additional training material. It includes not only the effect of the GSM transmission but also provides additional speaker information. Thus the performance of the SV system rises because more speaker information is available.

It seems that the adaptation with simulated GSM data is effective when the evaluation is also done with simulated GSM data. However when performing the evaluation with real GSM data, the adaptation can only capture a part of the difference between the fixed line data and the GSM data.

9.3.3 Results with simulated GSM recordings incorporating transmission errors

In our previous results we used our simplified simulation of the GSM transmission that disregards the effects of bit errors during radio transmission. To exemplify the results with the more elaborated GSM simulation we deploy the transcoded recording set 'sim GQ FR C-0.08' where a typical bit error rate of a radio link is used. We restrict our presentation here to the usage of GMM models in the adaptation scenario 'sim-real' together with the transformation set assignment 'per client'. This combination could be regarded as the most probable adaptation scheme for a real world application.

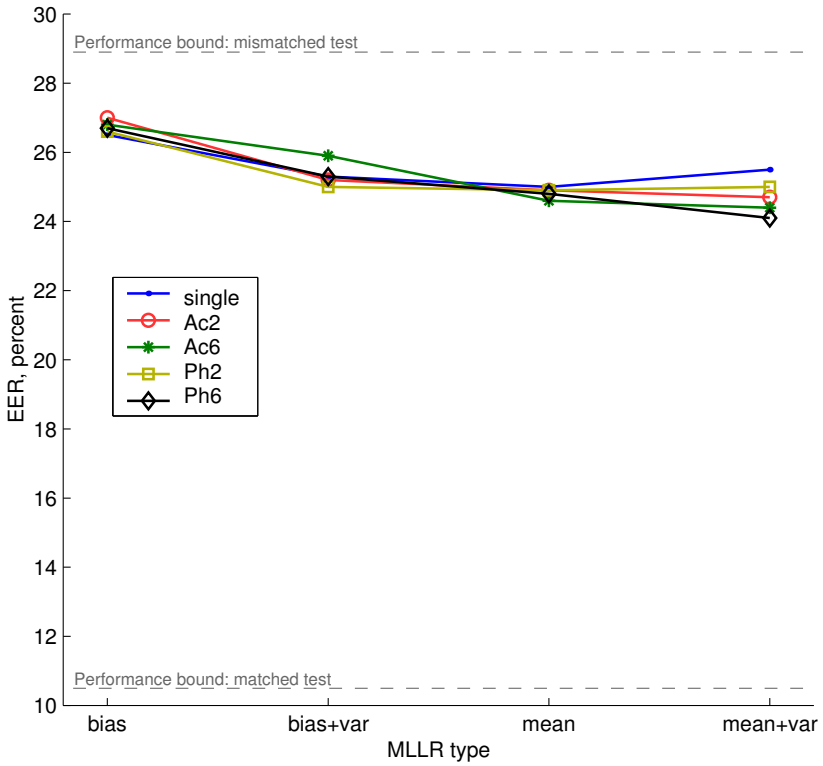

Figure 9.5: Dependence of the EER of the GMM SV system on the adaptation MLLR type (result plot type 1). Adaptation scenario 'sim-real'. The simulated GSM speech data is taken from the transcoded data set 'sim GQ FR C-0.08'. For clarity the transformation set assignment is fixed to 'per client' while the type of clustering and the number of regression classes is varied.

Figure 9.5 shows the EER in a result plot type 1 with a varying number of regression classes for both acoustic and phonetic clustering. By comparing these results with the corresponding ones from figure 9.2 we find only small differences. This comparison can also be done more easily with the data from table G.14 and G.15 given in the appendix.

It seems that generally the adaptations deploying MLLR types without variance scaling and adaptation data including a distorted channel (recording set 'sim GQ FR C-0.08') show a slight performance improvement. In maximum we achieve an EER improvement of 0.6% with the 'Bias' MLLR type and the clustering type 'Ph6'.

In turn, the MLLR types with variance scaling seem to perform better with the clean channel adaptation data 'sim GQ FR' compared to the distorted channel data. Adapting on distorted channel data, the greatest performance deterioration compared with the data from figure 9.2 is given by 0.7% for MLLR type 'Bias+VarScale', using again the clustering type 'Ph6'.

Please note that both measured differences are far below the statistically significant size of 2.8% with $\alpha = 0.95$. With these confidence intervals none of the differences between the experiments of figure 9.5 and those from figure 9.2 can be regarded as relevant. Using clean channel data, we achieve the best EER of 24.1% with a phonetic clustering with six classes and the MLLR type 'Mean+VarScale'.

We can state that the adaptation data using a distorted channel does not give any advantage. Using clean channel data shows the best result with an EER of 23.6% deploying an acoustic clustering with six classes and the MLLR type 'Full+VarScale'.

The results for the data sets 'sim GQ FR C-0.08' and 'sim GQ FR C-0.13' differ only in maximum by 0.2%. The detailed performance figures are again listed in appendix G, where the results for the former data sets are given in table G.15 and G.16 respectively.

To summarise: we can state that with the more elaborated simulation of the effects of the GSM transmission no statistical significant improvement of the adaptation performance can be achieved.

9.3.4 Summary for GMM models

Judging all adaptation experiments for the GMM models, we can state:

- While a close match between the adaptation data and the evaluation data ('sim-sim' scenario) allows to compensate the effects of the GSM coding completely, the real world scenario ('sim-real') allows only a partly compensation of the mismatch. We find, that only 29% of the performance gap between the lower and upper performance bound can be closed (MLLR type 'Full + VarScale', clustering 'Ac6' or 'Ph6', 'per client' set assignment).

- Deploying only a common transformation set gives a deteriorated performance, especially for the MLLR types involving a full transformation matrix.

- As long as the 'per client' set assignment is used, there is virtually no dependency on the number of regression classes or the type of clustering (either acoustic or phonetic).

- Incorporating more sophisticated effects of the GSM transmission into the adaptation data gives practically no benefit.

For a first conclusion we can argue that the adaptation with simulated GSM data compensates only a part of the effects within real GSM data. The achieved compensation is based mainly on speaker dependent effects due to the basic GSM speech coding which can not be generalized for a large number of speakers. Even the basic MLLR type 'Bias + VarScale' which involves only two transformation vectors achieves already performance values comparable to the best values we found here. Thus we might conclude that the primary effects introduced by the GSM coding are the channel effects (which are partly compensated by the bias vector) and the effects by additive noise (partly compensated by the variance scaling).

Since including transmission errors in the simulated GSM data does not show any benefit for the adaptation process, we will continue our further experiments with the simple adaptation data sets 'sim GQ'.

9.4 Results for subword HMMs

In general, speaker verification systems with subword HMM models show a higher performance compared to GMM-based SV systems of the same complexity. The GMMs from the previous section use 32 Gaussian mixtures while the subword HMMs use 97 Gaussian mixtures in total. We expect therefore a noticeable absolute performance improvement not only due to the difference in the model type but also due to the higher overall number of Gaussian mixtures. However, the relative performance differences among the various parameters of the adaptation system should be comparable to the GMM SV system. Only the phonetic clustering is computed differently: the subword HMM system uses the explicit phoneme clustering described in section 8.6.2 while the GMM-based system uses the implicit phoneme clustering from section 8.6.2.

Similarly to the GMM experiments we start with the evaluation of the adaptation scenario 'sim-sim' which gives a first impression of the adaptation performance and its dependence on various influencing variables.

9.4.1 Evaluation using simulated GSM data

Figure 9.6 depicts the SV performance using the adaptation scenario 'sim-sim' in a 'result plot type 1'.

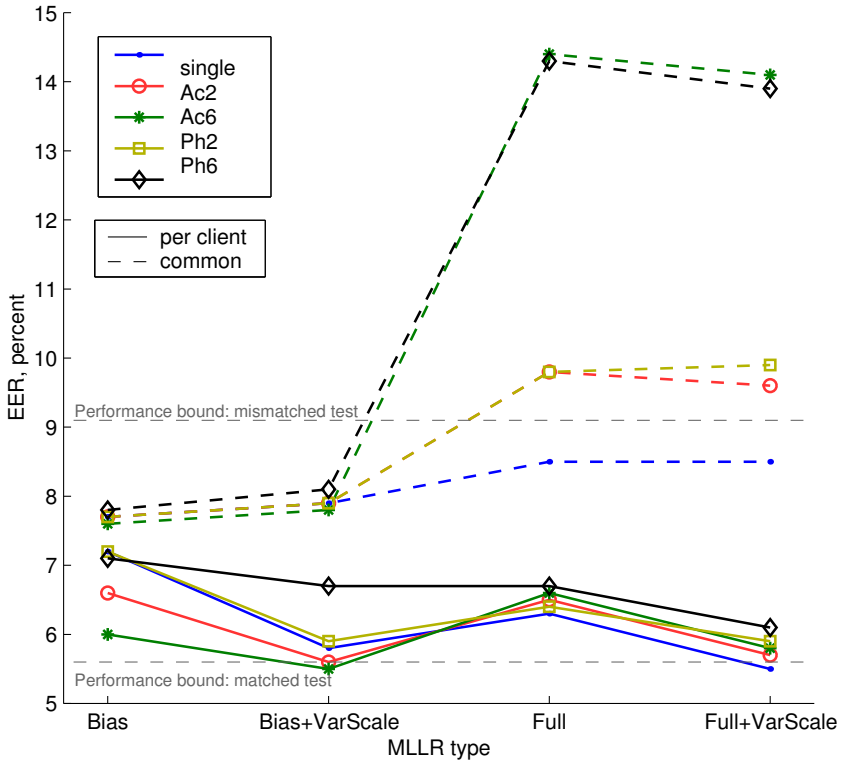

Figure 9.6: Dependence of the EER of the subword HMM SV system on the adaptation MLLR type (result plot type 1). Adaptation scenario 'sim-sim'. The type of clustering, the number of regression classes and the type of transformation set assignment is varied.

Generally, the adaptation schemes using the transformation set assignment 'per client' reach performances near the bound defined by the matched test. The best performance can be reached using the MLLR type 'Bias+VarScale' where all clustering types, except for the 'Ph6' clustering,

perform equally well. When comparing performance over all MLLR types, it seems that there is a consistent advantage of the acoustic clustering over the phonetic clustering. The fact that some adaptation performances fall even below the matched test bound has to be attributed to the statistical variations of the performance measure.

The common transformation shows again the familiar performance degradation for the MLLR type where the full mean transformation matrix is involved. For the remaining MLLR types, an EER improvement of about 1.0% can be achieved compared to the performance bound from the mismatched test case.

9.4.2 Evaluation using real GSM data

When applying the 'sim-real' adaptation scenario as depicted in figure 9.7, we find a similar pattern of the performance characteristics. The relative performance differences remain roughly the same compared to the 'sim-sim' scenario while the absolute values are shifted towards higher error rates.

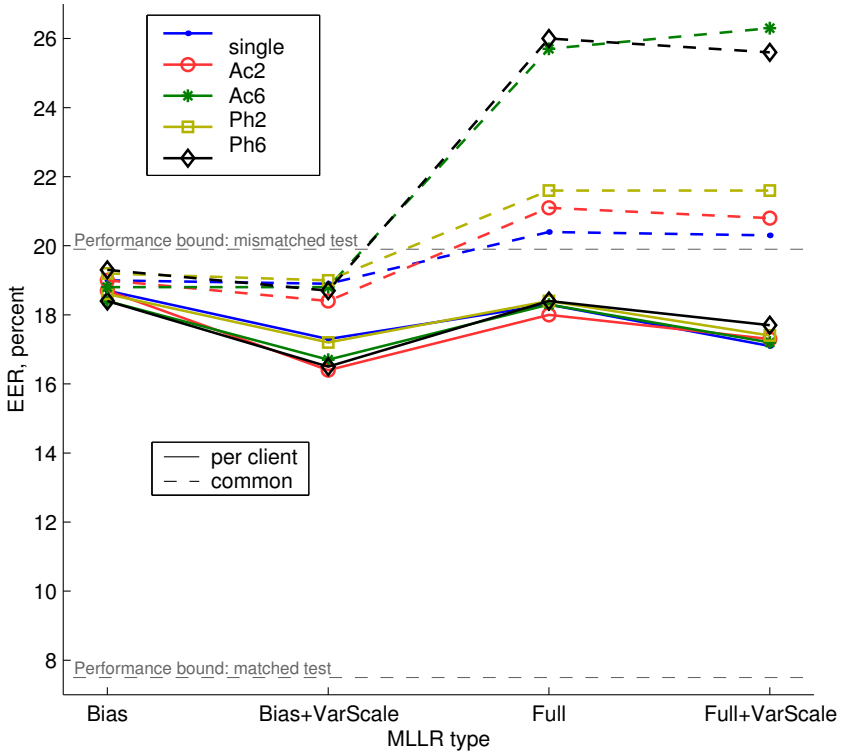

Figure 9.7: Dependence of the EER of the subword HMM SV system on the adaptation MLLR type (result plot type 1). Adaptation scenario 'sim-real'. The type of clustering, the number of regression classes and the type of transformation set assignment is varied.

When comparing figure 9.7 with figure 9.6, we find a similar pattern for the performance plots. Again the more detailed system which uses transformation sets 'per client' shows a higher performance (i.e. lower EER) than the system using a 'common' transformation set. When applying the former assignment type, the best performance values are achieved using the 'Bias + VarScale' MLLR type. Again we find a slight advantage of the acoustic clustering over the phonetic clustering. The optimum number of regression classes depends on the clustering types: the acoustic clustering works best when using two classes while the phonetic clustering shows better perfor-

mance using six classes. However these findings are not completely consistent when varying the MLLR type.

Using the common transformation set assignment we find again a large performance degradation for the MLLR types involving the full mean transformation matrix. In addition there seems to be the same dependency on the number of regression classes: the more regression classes the worse the performance.

To summarise our findings so far: we can achieve the best performance gain from 19.9% to 16.4% using an adaptation system with the 'per client' transformation set assignment, the clustering type 'Ac2' and the MLLR type 'Bias + VarScale'. Compared to the 'sim-sim' adaptation scenario, the performance does not reach the bound of 7.5% given by the matched test case.

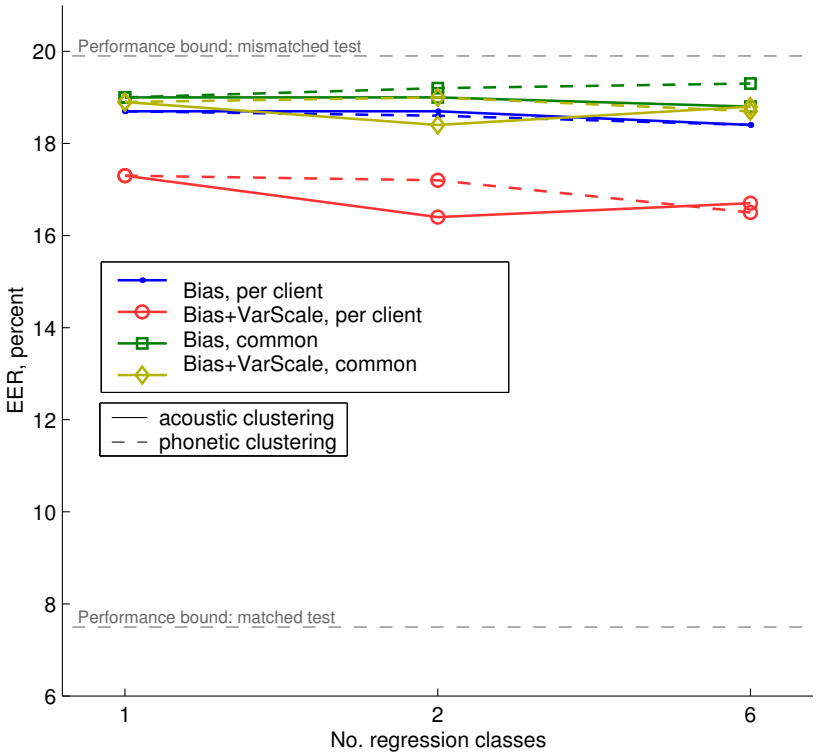

Figure 9.8: Dependence of the EER of the subword HMM SV system on the number of regression classes (result plot type 2). Adaptation scenario 'sim-real'. The type of clustering is varied for the MLLR type 'Bias' and 'Bias + VarScale'.

Figure 9.8 presents a type 2 plot of the MLLR types deploying bias vectors. The dependency of the performance on the number of regression classes and the clustering type is rather weak. Similar to the usage of GMM models (figure 9.4) the best performance improvement is achieved with the 'sim-real' scenario and the 'per client' set assignment. Only for the case of two classes we see that the acoustic clustering performs slightly better compared to the phonetic clustering.

Finally, we keep the MLLR type fixed by using'Bias + VarScale' and depict a 'type 2' plot in figure 9.9 with varying adaptation scenarios. Here we can compare the performance of the three practically relevant adaptation scenarios. Regarding the common set assignment, we see that using the real GSM data for adaptation gives a slightly better performance than using simulated GSM data. The best values are achieved using the clustering type 'Ac2' where the gain is 0.5% (18.5% with simulated GSM data vs. 18.0% with real GSM data). However, this difference is not significant.

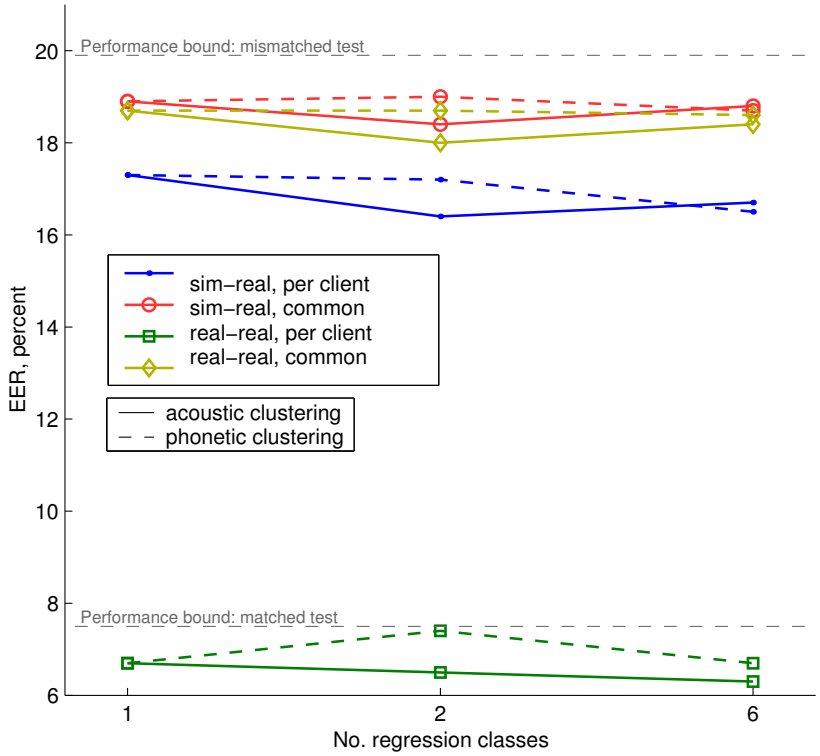

Figure 9.9: Dependence of the EER of the subword HMM SV system on the number of regression classes (result plot type 2). The type of clustering is fixed ('Bias + VarScale') while the type of adaptation scenario varies.

We can state that adaptation schemes in the 'real-real' scenario together with a common transformation set do not include speaker dependent or handset dependent properties of the real GSM recordings and therefore can not generalise these effects well enough.

In contrast, the 'sim-real' scenario together with the 'per client' set assignment performs clearly better compared to the 'sim-real' scenario deploying the common transformation set (clustering type 'Ac2'; EER = 16.4% vs. EER = 18.5) It seems that regarding speaker dependent effects in the GSM coding gives an advantage in the adaptation procedure.

Finally, the performance of the unrealistic adaptation scenario 'real-real' with the 'per client' set assignment reaches the performance bound given by the matched test case and even falls below this value. There is a dependency on the number of regression classes that we have already found with the GMM modelling. Again, the acoustic clustering gives a slightly better performance compared to the phonetic clustering.

9.4.3 Summary for the subword HMMs

We find very similar results for the subword HMM SV system compared to our results for the GMM-based SV system (see section 9.3.4). Again, only a small part of the performance gap between upper and lower performance bound (around 28%) can be bridged by the adaptation with the MLLR type 'Bias + VarScale', clustering type 'Ac2' and 'per client' transformation set assignment.

Similar to our previous results, the dependency on the number of regression classes and the type of clustering is very small for the 'per client' transformation set assignment. Although

for the subword-HMMs a different approach for the assignment of phonetic classes, the implicit assignment (section 8.6.2) is used, we do not find any statistical relevant influence of the regression class approach.

Since so far it turned out that the GSM coding effects seems to be highly depending on the client speaker, we further investigate this behaviour in the following section.

9.5 Differences in estimated MLLR bias vectors

Our goal in this section is to give more detailed insight into the performance differences between the adaptation systems using a 'per client' set assignment and those using the 'common' assignment. The adaptation scenarios 'sim-sim' from sections 9.4.1 and 9.3.1 are capable to reach the performance bound given by the matched test case. It seems that for simulated GSM data the channel adaptation can be performed successfully. While the 'per client' set assignment gives the best performance, the 'common' set can still generalise some of the channel effects.

We find a completely different picture when comparing both set assignment types using the 'real-real' adaptation scenario. Here, the adaptation with the 'common' type reaches a much worse performance compared to the adaptation deploying the 'per client' type.

Please note that we apply some adaptation schemes which are not practically relevant for a real adaptation system. However, they allow us to compare the generalization of the transformation sets deploying only simulated GSM data and those using real GSM data. At the end of section 9.3.2 we already stated that the 'real-real' scenario together with the 'per client' transformation set assignment inherently provides additional speaker information beside the channel information to the SV system. This fact results in a higher performance than the performance bound given by the matched test case. We will ignore this effect in the following and assume that the performance bound is not exceeded.

To simplify the investigations, we performed our analysis in this section only for the GMM SV system. Since here all transformation sets are applied to a single state, we can easily compare the differences between different transformation sets and normalize their values on the unadapted model parameters.

In order to compare the differences between a common transformation set and the client dependent transformation sets, we select the MLLR type 'Bias'. For each client speaker and each cepstral component i of the estimated bias vector we calculate the absolute value of the normalised difference between the bias value in the common case $b_i^{(common)}$ and the bias value of a client speaker $b_i^{(client)}$:

$$d_i^{(client)} = \frac{\left| b_i^{(common)} - b_i^{(client)} \right|}{\sigma_i^{(common)}} \tag{9.1}$$

where the normalisation is done using the standard deviation $\sigma_i^{(common)}$ for the cepstral component i estimated from the world speakers.

In other words, the difference $d_i^{(client)}$ gives the normalised distance of the bias vector component i between a transformation set estimated on the world speaker and a transformation set estimated on the client's data. The distance is measured in units of the general standard deviation of this cepstral component in the speaker population (calculated from the world speaker population).

High distances indicate, that the model of a particular speaker needs for a particular cepstral component different transformation parameters than the ones estimated in the common transformation set. Various sources of variation could cause this difference: the type of speech coding in

the cellular phone, a different type of interaction between the speaker's acoustic properties and the coding algorithm, different microphone characteristics, different type of background noises and varying transmission quality. In turn, a low distance $d_i^{(client)}$ indicates that the particular bias component can be estimated on a distinct speaker group and can be applied generally, regardless of the speaker identity. This would be the case when the effects of the speech codec are not speaker dependent and could be therefore eliminated by a common transformation set.

In the following plots, the distance is encoded by a color ranging from black (distance 0.0) to white (distance 0.5). Higher distances than 0.5 are limited to 0.5.

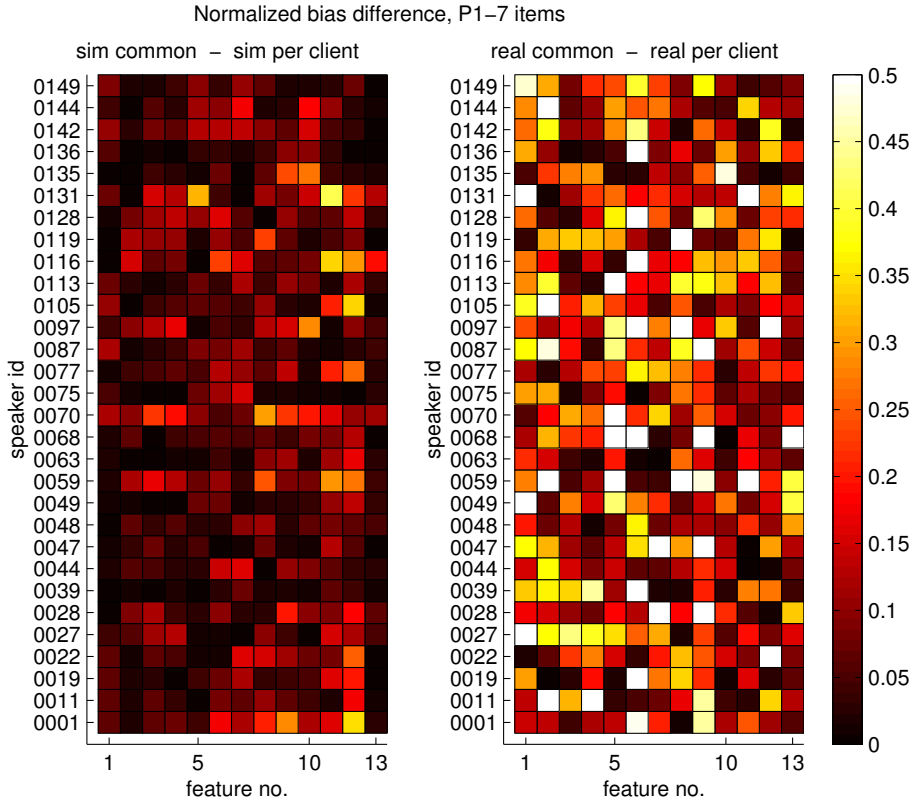

Figure 9.10: Normalised bias distances $d_i^{(client)}$ between bias vectors of the 'common' transformation sets and the 'per client' transformation set. GMM based SV system, adaptation MLLR type 'Bias', clustering 'single', adaptation scenario 'sim-sim' (left) and 'real-real' (right). Utterances taken from split 'S2' of the triplet subset. Only base cepstral features and energy are considered.

Figure 9.10 shows the normalised bias distance for each client speaker and each feature vector component. Indices one to twelve indicate the base cepstral features while the energy is listed as the 13th feature. On the left hand side, the adaptation scenario 'sim-sim' is used while on the right hand side the results for the adaptation scenario 'real-real' is shown. We can state for simulated GSM data that large bias differences exist only for a few client speakers. Further it seems that the higher order cepstral features show higher differences than the lower order ones.

The situation is quite different for the adaptation scenario 'real-real'. Nearly every speaker shows bias differences of more than 0.4 in one or more features. Compared to the previous case there seems to be no consistent pattern in the distribution of the differences among speakers or among the feature components.

The overall mean of all distances, where we average over all client speakers and all feature components, is denoted as $\widetilde{d}_i^{(client)}$; it is given in table 9.4.

	Average bias distance $\widetilde{d}_i^{(client)}$ using adaptation scenario...	
Item type	'sim-sim'	'real-real'
Triplets	0.07	0.22
F1	0.08	0.21

Table 9.4: Average bias distance between bias vectors of the 'common' transformation sets and the 'per client' transformation set. Same parameters used as in figure 9.10.

In order to investigate the effect of the linguistic content of the speakers' utterances on the adaptation performance we also set up a SV system using the fixed speech items F1. All further processing including the adaptation schemes resembles the processing used with the triplet speech items. The detailed bias differences are depicted in figure 9.11 while the average bias differences are given in the second line of 9.4. Again we can state that the bias differences for the simulated

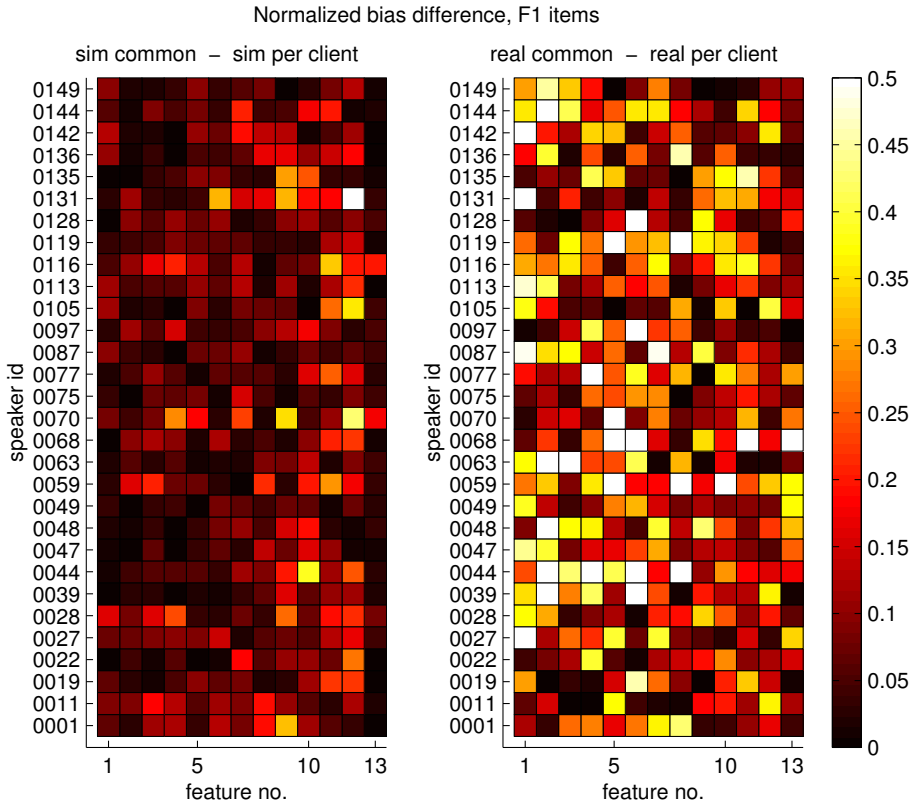

Figure 9.11: Normalised bias distances between bias vectors of the 'common' transformation sets and the 'per client' transformation set. GMM based SV system, adaptation MLLR type 'Bias', clustering 'single', adaptation scenario 'sim-sim' (left) and 'real-real' (right). Utterances taken from the F1 subset. Only base cepstral features and energy considered.

GSM data are much lower than the differences in case of using the real GSM data. The average distances are very close to the values received from the triplets data. Similar to the previous results there seems to be no consistent pattern in the differences when applying the real GSM data.

9.6 Summary

In this chapter we investigated the performance of our proposed adaptation approach for both SV systems based on the GMM and the subword-HMM models. The results match very closely between the two systems.

We found that for the real world application only a part of the performance gap, around 28%, between the matched and the mismatched case can be closed. Crucial impact on the performance of the adaptation scheme has the type of the transformation set assignment: only with individual transformation sets for each client speaker a reasonable adaptation performance can be achieved. We conclude that the effects of the GSM speech coding are speaker dependent and can not be generalized by a common transformation set that is applied for all client speakers.

The further analysis of the normalized bias distances from section 9.5 revealed that with real GSM speech a greater variability exists for the bias vectors between the client speakers compared with simulated GSM speech. Thus a common transformation set trained on the real GSM data can not capture the individual properties of each speaker and will show a low adaptation performance.

We found a small performance improvement by additional parameters, that were judged very promising in the beginning:

- The type and number of regression classes: Since it seemed in our preliminary experiments from section 7.5 that different phonetic classes are affected differently by the GSM speech codec, we would have expected that applying individual transformation sets to groups of Gaussian mixtures will compensate the nonlinear codec behaviour. However, it turned out that more regression classes do not lead to consistently better adaptation results and that the performance difference compared to a single regression class is clearly below statistically relevant differences.
- Inclusion of transmission errors into the simulated GSM data: Since the basic 'sim GQ' data does only include the effects of the GSM codec itself, we tried to incorporate the degrading of the speech signal due to transmission errors as well. Again, the adaptation performance is affected only subtly by the variation of the additional transmission channel.

Because both sophisticated approaches do not clearly contribute to the adaptation performance, we can easily fall back to a simple adaptation scheme without noticeable performance degradation. Regardless of the type of the SV system, we can use a single regression class, the 'per client' transformation set assignment and either the MLLR type 'Bias + VarScale' or 'Full + VarScale'.

We conclude, that besides the effect of the core GSM speech codec, other effects within the GSM recordings must be taken into account. Many additional parameters can have an impact on the recordings: the type (or even types) of GSM handset used by the client speakers, the acoustical environment causing additional noise in the recordings or changes in speaking style.

Since in the VeriDat corpus the choice of the current acoustic background for the recording was mainly left to the speakers (see section A.2), we can not separate these effects and investigate them isolated. Instead we can only refer to them globally as a single speaker dependent effect that needs further investigation.

The statistical relevance of the best achieved adaptation performances is discussed in the next chapter.

Chapter 10

Discussion

In order to get an overview of the performance gains of different adaptation schemes from chapter 8 we compare them with the achievable gains of simple retraining methods. Although we presented several approaches for channel compensation in section 8.1 we focus in our overview only on one compensation technique in the model space, our MLLR approach from section 8.3. Several other approaches from related work that we have presented in section 8.4 require a considerable higher effort for an implementation. We will try to compare our results with those of the publications from section 8.4 although many differences in the experimental setup do not allow a direct relation.

Our aim is to present the relative performance of the adaptation schemes compared to easy implementable retraining techniques. We include for this comparison a single iteration of the Baum-Welch reestimation using training and adaptation data similar to the one we used for the experiment "C-Base" in section 6.4.

10.1 Performance of adaptation and retraining techniques

Care was taken to render the experiments as much comparable as possible. However, due to subtle differences in each setup, we can not deploy exactly the same data sets for adaptation and retraining through out all six experiments in the comparison. We claim that despite these small differences the results can be compared as we tried in the selection process to choose data sets that provide a comparable distribution of the sub-words. Thus the influence of the linguistic content of the utterances should be kept at its minimum.

Common to all experiments is the same data set for evaluating the performance (set 'S2 GQ evaluate', see section 9.1). Because the training set of experiment 5 contains a genuine GSM recording session ('20') and the training set of experiment 6 contains two GSM recording sessions ('15' and '20'), these sessions are not part of the evaluation set 'S2 GQ evaluate' in order to avoid a methodological error known as "testing on the training set".

Figure 10.1 shows the performance gains while more detailed quantitative results together with information about the deployed data sets are listed in table 10.1.

As we have done already in chapter 9 we plot two performance bounds that spawn the range in which performance improvements can be achieved. The baseline experiment (experiment 1) marks the lower performance bound, from which we expect improvements. The mismatch between the SV system using 'FQ' trained client models together with 'GQ' evaluation data marks the worst performance in this overview (EER = 28.9%). In contrast, the upper performance

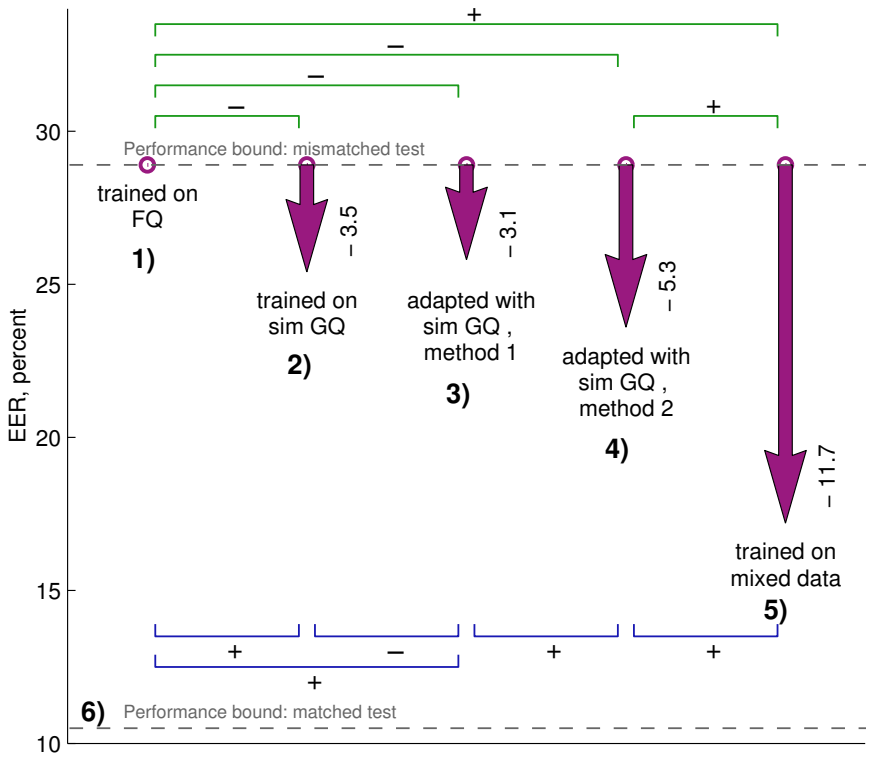

Figure 10.1: Performance gains for main types of training/adaptation schemes for environment adaptation of 'FQ' client models towards the 'GQ' environment. Baseline for all performance improvements are unadapted client models (1). Different training schemes (2 and 5) and different adaptation schemes (3 and 4) are compared (see text for more details). The upper performance bound (lowest EER) is given by client models trained on 'GQ' data (6). The performance differences can be judged by two types of statistical significance tests on the $p = 0.05$ error level; '+' denotes a significant difference while '-' denotes a non-significant difference. The values on the top indicate significance based on the confidence intervals calculated with the subsampling type 2 (see section 5.2); the values on the bottom are calculated using the binomial model (see page 29).

bound is given by the result of experiment 6. Here the lowest EER is achieved when client models are trained on 'GQ' data and thus matching test data is deployed for the evaluation.

Our main goal is to perform an adaptation of the system towards the 'GQ' environment using only the client's training data from the 'FQ' domain and a simulator of the GSM effects on speech data.

Experiment 2 deploys no adaptation of the speaker models but rather a retraining technique. The same data used later for adaptation ('sim GQ') is taken for a single Baum-Welch retraining step of the client models. Its performance of EER = 25.4% lies between the performances of the two adaptation schemes that we selected for our overview. The purpose of this experiment is to serve as a comparative performance value for the adaptation experiments. Since the retraining is less computational expensive than the calculation of the adaptation parameters, it might be an alternative in some cases.

10.1.1 MLLR adaptation with acoustic clustering

Experiment 3 and 4 are two selected adaptation experiments of the adaptation scenario 'sim-real' that we described already in section 9.3.2. Both are using the acoustical clustering scheme of

No.	Experiment	EER	EER confidence interval ($p = 0.05$)	
			based on subsampling type 2	based on rule of 30
1)	Mismatched test: training 'S2 FQ training', evaluation 'S2 GQ evaluate'; no adaptation	28.9	[21.1 36.7]	[27.0 30.8]
2)	Trained on 'sim GQ': training 'S2 sim GQ training', evaluation 'S2 GQ evaluate'; no adaptation	25.4	[18.3 32.5]	[23.9 27.0]
3)	Adapted with 'sim GQ', method 1: training 'S2 FQ training', adaptation 'S2 sim GQ training', evaluation 'S2 GQ evaluate'; clustering Ac6, Bias+VarScale, common transformation set	25.8	[18.4 33.2]	[24.0 27.6]
4)	Adapted with 'sim GQ', method 2: training 'S2 FQ training', adaptation 'S2 sim GQ training', evaluation 'S2 GQ evaluate'; clustering Ac6, Full+VarScale, per client transformation set	23.6	[16.9 30.3]	[21.8 25.3]
5)	Trained on mixed data (1 FG session, 1 GQ session): training 'S2 mixed training', evaluation 'S2 GQ evaluate'; no adaptation	17.2	[12.2 22.2]	[15.7 18.7]
5a)	Trained on mixed data (1 FG session, 1 GQ session): training 'S2 mixed training', evaluation 'S2 FQ evaluate'; no adaptation	6.9	[4.2 9.6]	[6.2 7.7]
6)	Matched test: training 'S2 GQ training', evaluation 'S2 GQ evaluate'; no adaptation	10.5	[5.5 16.0]	[9.4 11.7]

Table 10.1: Comparison of the main experiment types for adapting client models trained on the environment 'FQ' towards the environment 'GQ'. The experiment number refers to the one used in figure 10.1. Performance given in terms of the EER together with its confidence intervals based on the two approaches subsampling type 2 (see section 5.2) and rule of 30 (see page 29). The statistical significance of the performance difference is also depicted in figure 10.1.

the mixtures 'Ac6' that showed the best results in this particular combination together with the transformation set assignment (either 'common' or 'per client').

Experiment 3 is using a common transformation set assignment together with the MLLR type 'Bias+VarScale'. It can be regarded as a low complexity adaptation scheme because of its savings in the storage amount (common transformation set) and computational complexity (two single vectors of adaptation parameters). Its performance of 25.8% EER is only slightly below the performance of the retraining scheme from experiment 2 (25.4% EER). This is surprising as we would expect the retraining scheme performing much better: it computes a complete reestimation of both mean and variance parameters of the client models on an individual level

while the adaptation scheme from experiment 3 performs only global corrections using six bias vectors (for the mean vector) and six component-wise scaling vectors (for the variance vector).

The clustering of the mixtures using six regression classes can not be solely responsible for the nearly equal performance of experiment 3 with the retraining scheme (experiment 2): using no mixture clustering (scheme 'single') gives only a slightly deteriorated performance. Obviously, the higher complexity retraining scheme can be replaced by a common transformation set using a single bias vector and a single variance scaling vector.

In experiment 4 we use a more sophisticated setup of the adaptation system compared to experiment 3. Here we deploy a 'per client' assignment of the transformation sets and a full transformation matrix for the mean plus a variance vector scaling (MLLR type Full+VarScale). Compared with experiment 3, the performance improves by 2.2% towards an EER of 23.6%. The performance gain is linked with a greater complexity both for storage requirements (transformation sets stored per client, no offline computation before enrolment) and for the computational complexity (calculation of a transformation matrix plus the variance scaling vector).

When comparing the 'per client' adaptation from experiment 4 with the retraining technique from experiment 2, we find a performance gain of 1.8%. Although both techniques use the same amount of speech data, the adaptation technique achieves better results probably because the model parameters can not be retrained successfully from scarce speech data.

10.1.2 Training on mixed data

So far, performance gains could be achieved by using simulated GSM recordings that we derived from the fixed line recording sessions. Experiment 5 represents a different adaptation scheme by using mixed training data both from the fixed line and the GSM recording sessions. Again, to be comparable with the previous experiments, we select in total two recording sessions for the training here: a single session from the environment 'FQ' and a single session taken from the 'GQ' recordings. Table 10.1 provides more details on the training set denoted as 'S2 mixed training'.

This mixed training scheme provides the best performance gain and crosses the gap of the lower performance bound (experiment 1) and the upper performance bound (experiment 6) by more than the half towards 17.2% EER. However, we restrict ourselves to only two recording sessions per client and we assume, that no additional sessions for the adaptation are available except for transcoded data generated from the genuine recordings.

Due to reduction of the training material within both network types, the performance with fixed line data is degraded. Table 10.1 provides the EER using the 'FQ' evaluation data (experiment 5a). While all adaptation/retraining experiments show a fixed 'FQ' performance of 5.4% EER (see appendix G.1.2), the performance of the mixed data system degrades to 6.9% EER.

10.2 Statistical relevance of performance differences

In section 4.3 we presented several different approaches to judge performance differences in terms of the EER derived from the overall DET plot. It was shown (section 5.3) that the binomial model from page 29 and its approximation by the rule of 30 assumes that each client and impostor test is stochastically independent from each other and thus underestimates the variance of the DET plot. The two-stage sampling scheme for the genuine tests and the three-stage sampling scheme for the impostor tests, that we denote as subsampling type 2 in section 5.2, provides a much

more consistent match with the results of our bootstrap experiments. Generally, subsampling type 2 provides wider confidence intervals compared to the binomial confidence intervals.

The confidence intervals for the EER measured from the DET plot were computed using the procedure described in section 4.5. We will use both the binomial bounds and the subsampling type 2 bounds for the judgement of the statistical relevance the performance differences based on the error level $p = 0.05$. In figure 10.1 we have listed the relevance based on the binomial bounds on the bottom while the relevance based on subsampling type 2 is given on the top. We denote a significant performance difference of an experiment pair by a '+' while a '-' denotes a non-significant performance difference.

The subsampling type 2 bounds are generally much wider than the bounds based on the binomial approach, a property that we have already discussed in section 5.3.2. Only the larger differences between experiment pairs 1-5 and 4-5 can be taken for granted.

In contrast, the binomial confidence intervals are much tighter and render also smaller performance differences as statistical relevant. We find that only the difference between experiment 2 and 3 must be judged as a random result. Especially the difference between the results of experiment 2 (simple retraining) and 4 (adaptation with method 2) would be judged as statistically significant according to the binomial confidence bounds.

Our findings from section 5.4 revealed that the subsampling type 2 confidence intervals are supported by bootstrap experiments and might be more suitable for practical judgements.

Nevertheless, both methods for finding statistical relevant performance differences reveal that adaptation using simulated GSM recordings with a 'per client' assignment (experiment 4) gives a notable performance gain (5.3% absolute reduction of the EER) compared to the baseline system without adaptation (experiment 1). However, the much less sophisticated method of training with mixed data (experiment 5) shows again a clearly significant gain, roughly doubling the absolute reduction of the EER to 11.7%.

10.3 Comparison with related techniques

As we have mentioned already briefly in the introduction of this chapter, a meaningful comparison of our results with the results of related adaptation techniques from section 8.4 is difficult to accomplish. Despite the fact that the structure of many research systems for GMM based speaker verification use commonly deployed processing stages and techniques, many influencing factors remain that contribute to performance differences.

The acoustic and linguistic properties of the speech material both deployed for training and evaluation of the systems have a great impact. In addition differences in the size of the training data, the training procedures and the generation of normalisation stages like e.g. background models have a large influence on the final performance.

Thus a direct comparison of the performance figures for different systems is not meaningful. There remains only the possibility to compare the relative error reductions of different adaptation techniques with a baseline system without any adaptation applied. However, many publications do not report a baseline performance. While the papers of feature mapping [93] and speaker model synthesis [94] present performance figures for unadapted systems, the publications of the extended versions of the stochastic transform [95, 83, 81, 96, 82, 87, 86] present baseline performances where at least a CMS normalisation has been applied. In addition, no performance values for matching evaluation data is given although these figures would allow a judgement about the basic complexity of the recognition task with the deployed speech material.

Another deficiency of the cited publications is the investigated adaptation task. All publications perform adaptation to handset transducer effects. In [81, 83, 82, 96] the HTIMIT corpus and a GSM transcoded version of HTIMIT are deployed for investigations. Both corpora suffer from the fact that they were artificially generated under controlled conditions. The remaining publications from our selection used either only fixed telephone line speech data from NIST evaluation programs before 2004 ([86, 87]) or evaluated the adaptation on fixed line and cellular data separately.

Thus: none of the cited publications adapt HMM or GMM models to data from a different network type, namely cellular line data to fixed line data or vice versa.

10.3.1 Relative error reduction of related techniques

A meaningful comparison of the relative error reductions of different approaches requires to relate a performance difference to the totally achievable performance improvement (measured as the difference of the performance under mismatched and matched conditions). Since both performance bounds are given only in [82], we can report solely the relative performance reduction related to the performance in the unadapted (mismatched) case.

Since all cited publications use either speech data normalised with CMS or with RASTA-filtering ([93]), we select the related performance figures as baseline values. The relative error reductions are listed in table 10.2. Only results for cellular speech are reported.

Adaptation approach	**description in section**	**relative EER reduction**
Zero-th order SFT	8.4.1	16%
CSFT	8.4.2	16%
BSFT	8.4.3	19%
Feature mapping	8.4.4	17%

Table 10.2: Relative error reduction of different adaptation approaches (cellular speech data)

The relative error reduction of our best adaptation approach (experiment 4 in table 10.1) is computed to 29%. Please note that two facts deny a direct comparison of this figure with those in table 10.2:

- our baseline performance is measured without CMS normalisation.
- our adaptation performs adaptation to data from a different network type while the cited publications only perform adaptation to different channels within either fixed line or cellular networks.

10.3.2 Performance bounds for simulated GSM speech

An investigation similar to our adaptation scenario 'sim-real' can be found in [56]. Here, a text dependent speaker verification system deploying HMM word-based models is trained on both A-law and simulated GSM data. Contrary to expectations, the matched condition using GSM training and GSM evaluation yields a lower performance than the mismatched condition, where the training used A-law data and the evaluation is performed with GSM data.

For the LPCC features, a cepstral mean subtraction was involved. The maximum absolute performance difference between the mismatch evaluation and the matched evaluation is less than 0.5%; the maximum EER in this context is reported with about 2%. In contrast to these

values we find according to table G.1 a performance difference between mismatched and matched evaluation of 5% with the maximum EER (mismatched evaluation) of 9.1%. By regarding the range of the absolute error rates it seems that the VeriDat corpus contains more complex data. In fact, the training data in [56] comprises 112 digit words for the training of ten subword models while our 'S2' training scheme uses 84 uttered words to train 16 models.

Chapter 11

Summary and outlook

Summary

The previous chapters presented our contribution to environment adaptation of HMM-based speaker verification systems. In the first part of our work we dealt with the performance measures and the estimation of their confidence bounds. In addition we presented the results of preliminary experiments on the robustness of the MFCCs and LPCCs to mismatch either in the network type or in the level of background noise. The second part included a detailed analysis of the effects of the GSM codec on the LPCC speech features and presented an adaptation scheme in order to transform the client models of a speaker verification system trained on fixed line telephone data to application with GSM network data.

For the first part, chapters 5 and 6 are mainly relevant for our summary. We start with the experimental evaluation of different approaches to estimate the variability of the false rejection rate (FRR), the false acceptance rate (FAR) and the equal error rate (EER). Many publications report differences for these error rates in the range of one-tenth of a percent when comparing results of their experiments. Very often, no indication is given if and how these differences can be judged as statistically relevant. Therefore, one of our main intention has been to select a reliable way to estimate the confidence bounds of these error rates for our own experiments.

Several authors ([5, 40, 46, 47, 103]) express the need for estimating the confidence bounds in order to interpret reliably the success of performance improvement techniques. The only resource that addresses the confidence intervals of the FRR and FAR thoroughly can be found in [5] (see also page 31). However, the origin of the estimation formula for the FAR (denoted here as 'Bickel's formula') remains unclear and, even worse, it applies only to cross-comparison tests where every speaker acts once as client speaker and once as impostor speaker.

We have postulated that the variability of the FAR can be modeled similar to the two-stage sampling approach which is already used by the variability model of the FRR in [5]. Similar to a related investigation in [8], we have shown in section 5.1 that there exist clear speaker effects in the score population. For the population of the pooled genuine speaker scores, we find that the client speakers contribute in varying degree to the global FRR. The same effects are found for the impostor speaker scores which determine the FAR. Here, the score distribution depends both on the identity of the client speaker model and the impostor speaker.

Our investigations revealed that the binomial approach for estimating the confidence intervals of the performance parameters is not valid for our data since the score populations are not independent identically distributed. Further on, we find that our assumption of a three-stage sampling scheme for the impostor speaker scores is valid.

The bootstrap approach allowed us to experimentally determine the confidence bounds of our baseline system. We have compared these empirical bounds with the predicted bounds given by the binomial approach, Bickel's approach (denoted here as 'subsampling type 1') and our estimation formula (denoted 'subsampling type 2'). We find clearly, that the binomial bounds underestimate the real variability of the DET plot. Our main result for this chapter is that our proposed approach 'subsampling type 2' matches better with the empirical bounds compared to bounds of the 'subsampling type 1' approach. Especially in the lower FAR area, the 'subsampling type 1' bounds tend to overestimate the empirical confidence bounds. We decided to use the confidence bounds based on our 'subsampling type 2' approach to judge the performance differences of our following experiments.

Typically for the EER, we estimate single-sided confidence intervals on the $p = 0.05$ error level of 5-6%. Thus, only performance differences in this range can be judged as statistically relevant. In order to increase the resolution, our database should contain more speakers rather than a large amount of utterances per single speaker. Further data collections should respect this requirement.

In chapter 6 we have introduced a robustness coefficient which allows to express performance differences of the speaker verification system on disjoint data subsets with a single measure. Totally robust systems to different mismatched test sets have a relative robustness coefficient of $rc_{\mathcal{A}} = 1.0$ while typically systems in real world applications reach only values below unity. The definition can be extended to an arbitrary number of data subsets and can include additional weighting of the subset relevances.

We apply the new measure in our preliminary experiments on the robustness of the LPCC and MFCC feature sets with regard to variation of the network type and the level of background noise. The LPCCs turn out to be highly robust to variation of the noise level. Since the absolute performance of both feature sets are nearly identical and also in other publications [56, 57] no clear advantage of one feature set over the other can be found, we selected the LPCC features for our further experiments.

Our primary motivation for the second part of our work was to improve the total robustness of the LPCC features by improving their susceptibility to variations in the network domain. The aim was to design an adaptation scheme that transforms the speaker models of a speaker verification systems trained on fixed line telephone speech to evaluation with GSM network recordings. Our intention was to get by with a single-session enrolment containing seven utterances of the number triplet items. Additional data for estimating the transformation parameters should either be calculated from GSM-transcoded enrolment sessions or from real GSM recordings of an off-line population, the world speakers. In the first case, the transformation can be calculated for each client ('per client' assignment) or for all client speakers in common ('common' assignment). For practical reasons, the second case allows only the 'common' assignment.

The core adaptation scheme is based on the well known MLLR algorithm. Different variants have been investigated including variations in the complexity of the transformations (bias vector, full transformation matrix, optional variance scaling), the number of regression classes and the type of regression classes (based on phonetically motivated similarity or acoustically motivated similarity).

First experiments compared original fixed line recordings with their counterparts that have been transcoded by the GSM algorithm. The comparison took place by calculating the Bhattacharyya distance of the first twelve LPCC coefficients between corresponding phonetic segments. Compared to the original recordings, deviations were mainly found in the simulated GSM recordings for voiceless segments (fricatives and plosives). The same effects can be found in real GSM recordings. The voiceless phonemes might suffer from the residuum compression stage in the GSM codec which is responsible for a large part of the data reduction. The dependency on

the phoneme type was taken as a primary motivation for using regression classes in the MLLR algorithm: different classes of speech segments could then be transformed individually by the adaptation system.

It turned out that the simulated recordings seem to incorporate only a part of the effects that we find for real GSM data. Additional effects such as the different and possibly more noisy recording environment, varying transmission quality and differences in the handsets are responsible for further signal degradation. In order to include at least the transmission effects into our simulated GSM recordings, we extended our codec simulation with the channel coding stage and injected bursty errors with typical error rates.

Chapter 8 presented our proposed adaptation scheme and compared it to related work from literature, namely the stochastic feature transformation (SFT) together with its various extensions [71, 95, 87, 86] and the feature mapping (FM) approach [93]. A detailed comparison of the performance values is not possible since the underlying databases and the experimental setups differ to a large extent. However, all reported results share the common property that only the mismatch in the transducer type or the handset type is targeted by the adaptation systems. No cross comparison of fixed line and cellular data is reported. This fact is also expressed by rather low equal error rates of around 13% in the mismatched test case and 10% in the matched test case [93].

In Chapter 9 and 10 we presented in detail the results of our adaptation system and compared them with standard retraining methods. For the GMM SV system, we find an EER of around 11% in the matched test case (trained and evaluated with real GSM data) and an EER of around 29% in the mismatched test case (trained with fixed line data, evaluated with real GSM data). Our best adaptation system using six regression classes (defined by acoustical similarity) together with the transformation type 'Full+VarScale' (full transformation matrix for the means and variance scaling) and the transformation set assignment 'per client' reduces the EER by roughly 5% to around 24%. The adaptation data was calculated from the single enrolment session (seven number triplets, around 28 seconds speech in total) which was transcoded by the simulated GSM codec. In contrast, a complete retraining of the speaker models by using the simulated GSM speech achieved an EER of around 25%.

According to the confidence intervals based on our 'subsampling type 2' approach, these performance differences can not be judged as statistically significant on the $p = 0.05$ error level. However, when including the results of all our adaptation experiments, we can note a general trend that only a part (around 28%) of the performance gap between the matched and the mismatched case can be closed with adaptation using simulated GSM data.

Our results indicate that the adaptation performance is mainly based on the adaptation scenario: using simulated GSM recordings in order to compute transformation sets for each client speaker (scenario 'sim-real' with 'per client' transformation set assignment) achieves generally the best performance. To a smaller part, the complexity of the transformation is relevant for the performance. In most cases, using a bias for the mean vector and a variance scaling vector gives a sufficient adaptation performance. The number of regression classes and also their type (phonetical or acoustical) does not show a significant impact. Also the inclusion of transmission effects in the simulated GSM recordings, which should render them more realistic, did not improve the adaptation performance significantly (improvement of the EER below 1%).

Our analysis from section 9.5 revealed that real GSM speech contains speaker-dependent effects that can not be generalised by a common tranformation set which is computed from real GSM data of the complete speaker population. The remaining differences might be based on the speaker characteristics themselves or indirectly on other speaker-related properties such as the type of cellular phone used for the recordings or the selected recording environment. Since

the VeriDat database does not provide further information in this area, the true origin of the differences can not be investigated.

Outlook

Although many advances in speaker recognition have been made, the problems with variability, especially in the channel and recordings conditions, still remain. For our own contribution to this field, we find several topics that can be a starting point for further work.

We have seen that transformation sets calculated from real GSM data are largly speaker dependent and can not be generally used for a successful adaptation of speaker models. When keeping the constraint of using only a single enrolment session from the fixed telephone network, the transformation sets based on real GSM data must be calculated from an off-line speaker population. Future work could investigate if the data from this population could be grouped using different criteria either for the speakers (e.g. gender, age) or for individual recordings (information on phone type, automatic classification of recording quality). The resulting collection of different transformation sets might be used to adapt the speaker models individually to the current conditions.

The usage of a stereo database which comprises simultaneous recordings from a fixed telephone and a GSM handset might reveal further effects in GSM recordings. Changes in the speaking habits and the influence of the acoustic background might be separated from the influence of the GSM transmission itself. However, the design of the data collection must ensure that realistic recording conditions are used that match with later applications.

A possible benefit for the recognition performance might be gained if GSM telephone calls would incorporate additional quality measures about the current radio link, that can be exploited by any speech processing system. The recent introduction of the broadband UMTS technology might also help to promote speaker recognition applications if dedicated protocols allow either transmission of high-quality speech or of feature streams that have been already captured inside the cellular phone.

Focusing on the acoustical mismatch alone might not deploy all possibilities for a successful adaptation. Recent research introduced different score normalisation techniques that rely on a (pseudo) impostor score distribution: Z-norm, H-norm, T-norm and many other [104]. Current state-of-the-art speaker verification systems deploy very often one of these score normalisation techniques together with other normalisations in the feature or model domain. Future work might reveal which of these techniques matches best with our adaptation approach.

Current speaker verification systems still lack the robustness for applications that could be accessed via cellular phone calls. Advances not only in the signal processing and classification techniques but also in acoustic-phonetics, voice perception and psycho-acoustics will be necessary for future systems in order to operate reliably in varying environments.

Appendix A

The VeriDat Database

The VeriDat database was created by T-Nova, Deutsche Telekom Innovationsgesellschaft mbH, and the Bavarian Archive for Speech Signals (BAS) as a database for German speaker verification in fixed and mobile telephone networks. It is an extension of the standardized specification for speaker verification databases as published in the SpeechDat project [89]. VeriDat contains an additional set of 19 recording items including number triplets and spontaneous speech aside from the 21 items defined in the SpeechDat specification.

The main idea of VeriDat was to create a resource suitable for all kinds of speaker verification systems that covers the whole range of German dialects and uses different recording environments (quiet and noisy background) as well as different networks (fixed network and cellular phones).

The database comprises 7GB of speech material and is distributed on two DVD-Rs containing all signal and label files, support files and documentation. The speech material is stored according to the European ISDN standard as 8bit, 8kHz, A-law encoded data. The label files are formatted according to SAM [105] and contain recording and speaker information, the prompted text, the transcription and a small set of noise markers.

Each of the 150 speakers were recorded in 20 sessions with a minimum break of three days between the sessions. The recordings took place mainly in autumn and winter 1999/2000. Additional sessions were recorded to replace corrupt material found in the full-cover validation resulting in a nearly 100% error free database. The SpeechDat compatible part of the database has been validated successfully by the Speech Processing EXpertise center (SPEX) in Nijmegen, Netherlands[1].

A.1 Speaker population

The recorded speakers closely represent the German population with respect to the distribution of German accents (13 dialects plus one bin for foreign accent). The gender distribution within each accent group and within the five age groups is perfectly balanced. Some of the participating speakers are related to each other and their relationship (brothers, twins, etc.) is documented. One pair of identical twins is among the related speakers. The VeriDat documentation contains a complete description about the distribution of the speakers‘ properties; the key properties are listed in appendix B.

[1] www.spex.nl

A.2 Recording conditions

The recordings can be divided by two sets of criteria: the recording environment (labeled "Quiet" or "Noisy" in the following text) and the network called from (labeled "Fixed" or "GSM"). The recording protocol defined for all speakers which session had to be recorded in which environment and from which network (see table A.1).

recording condition	session list
"Fixed/Quiet"	'01','03','05','09','12','13','17'
"Fixed/Noisy"	'07','14','19'
"GSM/Quiet"	'02','06','08','11','15','18','20'
"GSM/Noisy"	'04','10','16'

Table A.1: Mapping between environment/network and recording sessions.

The ratio of quiet to noisy sessions is independent of the partitioning in networks and vice versa (see table A.2).

	Fixed	GSM	$\sum$
Quiet	7	7	14
Noisy	3	3	6
$\sum$	10	10	20

Table A.2: Partitioning of recording sessions per speaker with regard to environment and network.

There had been no restrictions on the handsets used. Telephones using DECT technique but connected to ordinary fixed networks were treated as regular fixed network connections. The judgment about the degree of noise or quietness was left to the speakers though they were instructed by simple rules and sample recordings. This leads to a great variation of noise especially in the "Noisy" part ranging from static noise to loud cross talk (e.g. a small boy in the background whining about his father doing a telephone call).

A.3 Speech items

General comments

The VeriDat database contains 40 recording items per session and speaker among them sequences of digits with various length, number triplets, application words, spelled names and words, names of people, phonetically rich sentences and spontaneous responses to questions e.g. concerning the current recording location. Some items have been both printed on the prompt sheet and text-prompted before the recording to ensure the desired pronunciation of the digit strings.

There is another distinction between items that are speaker-specific e.g. the phonetically rich sentences and items fixed for all speakers e.g. the number triplets. The latter group uses the same mapping of prompts to the recording sessions for every speaker.

All items within a session are identified by a code of one letter followed by a digit. The first number triplet is e.g. identified by the code 'P1'.

Number triplets

In our experiments we decided to use the number triplets of two-digit numbers (e.g. "21 35 76", spoken as "ein-und-zwanzig fünf-und-dreissig sechs-und-siebzig"). Each session contains seven recordings of these triplets taken from a set of 140 triplets in total. Their identifier code 'P1' to 'P7'. The length of a single utterance is around four seconds.

The set of triplets is derived from the YOHO database [106] which defines 136 items and it is extended by four additional triplets at the end. However the different session structure between YOHO and VeriDat causes problems when selecting single sessions from the VeriDat database:

- YOHO uses 4 enrolment sessions with 24 triplets each and 10 test sessions with 4 triplets each. In each session the distribution of the different sub-words of the numbers ("twenty", "one", ...) are designed to be close to equal.
- VeriDat does not have predefined sessions for enrolment and test. All sessions contain seven triplets which are derived from the order given by the YOHO database. Thus we receive a non-equal distribution of the number words.

In the case of text-dependent speaker verification (using individual sub-word models) selecting a single VeriDat session for enrolment will lead in most cases to missing observations of some sub-words. In consequence the associated models can not be trained using only material from the client speaker. A more balanced distribution can be achieved by combining speech data from several sessions.

A.4 Other databases

A decade ago, speaker recognition had been considered as a marginal field of speech technology. Only few public databases were available on this topic. Among the most often cited databases are TIMIT/NTIMIT, the NIST speaker recognition evaluation databases, KING, YOHO for English databases and POLYVAR as a French database.

Increasing interest to speaker verification lead in the last years to the development of several especially non English corpora. The European project COST250 "Speaker Recognition in Telephony" promoted the realization of the speech database POLYCOST for 17 European languages. Beside these activities several smaller databases were created especially for French (LIMSI/CNET) and Spanish (Ahumada).

Several detailed overviews of currently publicly available databases for speaker recognition/verification can be found in [107], [108] and [109].

Most of the databases are multi-session databases that capture temporal intra-speaker variability. Several of them contain recordings from a speaker over a time span of three or more months. This is motivated by an investigation of Furui [110] that showed that a speaker's feature subspace increases during the first three months of measurements and stays then relatively constant.

Many of the multi-session databases contain variability of the telephone handset used and the caller environment.

In order to investigate intra-speaker variability and also inter-speaker variability, many databases (SpeechDat, PolyVar, LIMSI/CNET) use a split speaker population. The first set of speakers provides a small number of clients with many sessions while the second set provides many impostor speakers with only a single session.

Mimicry has been included to our knowledge only in the CSLU database [111], where in each call the speaker is asked to imitate a given prompt phrase. There is a need to included utterances from deliberate impostors into databases as almost all experiments in literature use non-genuine utterances with normal speaking style and thus, only the effects of unintended mimicry are captured.

The probably two most popular databases are the TIMIT with its various derivatives (NTIMIT, CTIMIT, HTIMIT) and the yearly NIST databases.

A.4.1 TIMIT,NTIMIT, CTIMIT, HTIMIT

The original TIMIT family is poorly suited for evaluation purposes of speaker recognition and speaker verification systems primarily due to its unrealistical recording design. There is no intersession variability as each speaker is recorded in a single session. The originally wideband recordings in a sound booth (TIMIT) were later rendered more realistically by playing back transmitting the original recordings through artificial mouths over telephone networks (NTIMIT), over cellular networks (CTIMIT) or through 10 different telephone handsets (HTIMIT), including carbon button and electret microphones, and a high-quality Sennheiser head-mount microphone.

A.4.2 NIST Speaker Recognition Evaluation

Many publications on speaker recognition and verification have their origin in the contribution to the yearly ongoing NIST speaker recognition evaluation program [55]. Since 1996 this program aims at creating a special focus on text-independent speaker recognition and to support the development in this topic. Among the notion *speaker recognition* are one-speaker detection (equals to speaker verification), two-speaker detection, speaker tracking and speaker segmentation.

The evaluation program is open to all researchers and is typically concluded by a special workshop for summarising of the results of the participating research groups. Its activities include a common evaluation scheme for all participating partners which ensures easy comparison of different approaches and systems. The data provided for training and development and the final evaluation data is used very often in different publications as well.

Typically a training amount of two minutes is provided for each speaker, while test utterances have a length between 15 - 45 seconds. Most recordings were derived from the various Switch-Board projects which contained mostly fixed line phone data. In order to keep the results of different groups comparable, the EER is computed from the DET plots with given cost parameter ([39], see description in section 4.2.2 and 4.2.4): $C_{FR} = 10$, $C_{FR} = 1$ and $P_{client} = 0.01$.

The evaluation prior to 2004 did include various types data including non-English recordings and special sets for cellular phone data. However, no cross-comparisons e.g. the evaluation of the performance under channel mismatch (fixed telephone line phone vs. cellular phone) were included in the evaluation plans. The channel effects of various types of fixed telephone line handhelds including carbon button and electret microphones have been in the focus.

Since 2004 the focus shifted to deploy a mixture of both fixed telephone line and cellular data. The so-called *mixer corpus* includes recordings of both types and allows studying the effects of mismatched conditions. Also extended are the possibilities to explore the effects of the amount of training or evaluation data: limited data as short as 10 seconds is included as well as extended data with speech up to five minutes for studying the influence of long-term features.

Appendix B

Speaker data

speaker	sex	age	age group	accent	relation type	relation group
				Client set		
0001	M	61	5	ST	father-child	16
0011	F	7	1	ST	brothers-and-sisters	3
0019	F	27	2	MV		
0022	M	11	1	MV	mother-child	9
0027	F	63	5	BY		
0028	F	55	4	BY		
0039	F	43	3	BY	mother-child	13
0044	M	56	4	NI		
0047	M	28	2	NI		
0048	F	34	3	NI		
0049	M	32	3	NI		
0059	M	29	2	NW	father-child	18
0063	M	27	2	NW		
0068	F	7	1	HE	father-child	19
0070	F	18	2	HE		
0075	F	34	3	BY		
0077	M	24	2	SH		
0087	M	12	1	BB	father-child	31
0097	F	66	5	SA		
0105	F	39	3	NW		
0113	F	43	3	SA	mother-child	30
0116	M	30	2	BW		
0119	M	14	1	BW	father-child	26
0128	F	30	2	NW		
0131	F	22	2	BW		
0135	F	18	2	BB	twins	1
0136	F	40	3	HE		
0142	M	12	1	SH		
0144	F	27	2	BW		
0149	M	10	1	NW	mother-child	27
Total: 17 female speakers, 13 male speakers						
				Impostor set		
0003	F	18	2	ST	mother-child	12
0004	M	16	2	ST	father-child	17
0005	F	39	3	ST	father-child	16
0006	M	42	3	ST	father-child	17
0007	F	26	2	D		
0008	F	33	3	ST	mother-child	10

Table B.1: Speaker sets. Continued on next page.

speaker	sex	age	age group	accent	relation type	relation group
0010	M	13	1	ST	brothers-and-sisters	3
0012	M	30	2	MV	brothers-and-sisters	8
0014	F	39	3	NW	mother-child	27
0015	F	37	3	MV	mother-child	9
0016	F	35	3	MV	mother-child	11
0017	F	11	1	MV	mother-child	11
0018	M	39	3	SL	brothers-and-sisters	5
0024	M	33	3	BW		
0026	F	33	3	BW		
0031	F	24	2	BY		
0036	M	24	2	BY		
0038	F	23	2	BY	mother-child	13
0042	M	19	2	RP		
0046	M	56	4	NI		
0050	F	23	2	NI		
0052	M	67	5	TH	father-child	15
0053	F	64	5	TH	mother-child	14
0054	M	36	3	TH	father-child	15
0057	F	56	4	TH		
0058	M	61	5	NW	father-child	18
0062	M	30	2	NW		
0065	F	25	2	NW		
0066	M	35	3	HE	father-child	19
0067	M	10	1	HE	father-child	19
0069	F	23	2	HE		
0072	F	11	1	HE		
0073	M	60	4	BY		
0076	M	43	3	NW		
0078	M	28	2	SH		
0082	F	15	1	SA	brothers-and-sisters	4
0083	M	55	4	BY	father-child	29
0084	F	25	2	BY	father-child	29
0085	M	46	4	BB	father-child	31
0086	M	9	1	BB	father-child	31
0088	F	11	1	BB	mother-child	10
0090	F	44	3	HE		
0092	M	58	4	SH		
0094	M	13	1	BB		
0098	F	12	1	SA	brothers-and-sisters	4
0099	M	8	1	SA	brothers-and-sisters	4
0101	F	30	2	SH		
0102	M	29	2	SA		
0118	M	26	2	D		
0120	M	35	3	SL	brothers-and-sisters	5
0121	F	16	2	BW	father-child	26
0124	F	39	3	BW	mother-child	25
0126	F	51	4	NW		
0129	F	51	4	OTHER	mother-child	23
0130	F	24	2	NW	mother-child	23
0134	F	18	2	BB	twins	1
0139	M	53	4	NI		
0143	M	34	3	NW		
0147	M	26	2	BW		
0148	F	13	1	NW	mother-child	27
Total: 30 female speakers, 30 male speakers						
World set						
0021	F	10	1	NW		
0023	M	59	4	BW		

Table B.1: Speaker sets. Continued on next page.

speaker	sex	age	age group	accent	relation type	relation group
0025	M	35	3	BW		
0030	M	30	2	BY		
0032	M	27	2	BY		
0034	M	29	2	BY		
0041	F	31	3	RP		
0043	M	56	4	NI		
0045	F	50	4	NI		
0055	F	36	3	TH		
0064	M	29	2	NW		
0071	M	10	1	HE		
0079	M	26	2	SH		
0089	F	19	2	SH		
0091	F	51	4	OTHER		
0095	F	35	3	BB		
0096	M	67	5	SA		
0104	M	23	2	BB		
0106	M	17	2	BW		
0108	M	29	2	SA		
0110	M	36	3	D		
0111	F	45	3	BB		
0114	M	27	2	SH		
0115	F	13	1	RP		
0123	M	21	2	NI		
0125	F	59	4	NW		
0140	M	21	2	RP		
0141	F	56	4	SH		
0145	F	18	2	BW		
0146	M	13	1	NW		
Total: 12 female speakers, 18 male speakers						
Development set						
0002	M	42	3	NW		
0009	M	25	2	ST		
0013	F	27	2	MV		
0020	M	31	3	MV		
0029	F	33	3	BY		
0033	M	32	3	BY		
0035	M	22	2	BY		
0037	F	25	2	BY		
0040	F	12	1	BY		
0051	F	44	3	BW		
0056	F	19	2	TH		
0060	F	40	3	NW		
0061	F	31	3	NW		
0074	M	24	2	NW		
0080	F	43	3	SH		
0081	M	16	2	OTHER		
0093	M	14	1	SA		
0100	F	54	4	RP		
0103	F	26	2	BB		
0107	M	49	4	SA		
0109	M	42	3	BY		
0112	M	47	4	BB		
0117	F	54	4	NI		
0122	F	54	4	BW		
0127	F	45	3	NW		
0132	M	55	4	RP		
0133	F	55	4	BB		
0137	F	28	2	TH		

Table B.1: Speaker sets. Continued on next page.

speaker	sex	age	age group	accent	relation type	relation group
0138	M	30	2	NW		
0150	M	21	2	NW		

Total: 16 female speakers, 14 male speakers

Table B.1: Speaker sets. Abbreviations used for accents: BY = Bayern; NI = Niedersachsen; ST = Sachsen-Anhalt; SH = Schleswig-Holstein; HE = Hessen; NW = Nordrhein-Westfalen; BW = Baden-Württemberg; BB = Brandenburg; MV = Mecklenburg-Vorpommern; TH = Thüringen; SL = Saarland; SA = Sachsen; RP = Rheinland-Pfalz; D = Hochdeutsch

age group	age
1	< 16
2	< 31
3	< 46
4	< 61
5	≥ 61

Table B.2: Definition of age groups

Appendix C

Feature sets

The HTK parameters of the basic feature sets LPCC_SD and MFCC_SD are given in the following table. The extended versions with CMS applied (sets denoted by LPCC_SD_CMS and MFCC_SD_CMS) are simply derived from the basic parameters by changing the parameter `TargetKind` to `LPCEPSTRA_E_D_A_Z` and `MFCC_E_D_A_Z` respectively.

Parameter	**LPCC_SD**	**MFCC_SD**
SourceKind	Waveform	
SourceFormat	Wave	
SourceRate	1250	
TargetKind	LPCEPSTRA_E_D_A	MFCC_E_D_A
TargetRate	100000	
PreEmCoef	0.97	
WindowSize	250000	
UseHamming	True	
LoFreq	N.A.	300
HiFreq	N.A.	3400
LPCOrder	14	N.A.
NumChans	N.A.	24
NumCeps	12	
CepLifter	0	
ENormalise	False	
UseSilDet	True	
MeasureSil	True	
SpeechThresh	9.0	
SilEnergy	0.0	
SpcSeqCount	10	
SpcGlchCount	0	
SilSeqCount	100	
SilGlchCount	2	
SilMargin	20	

Table C.1: HTK parameters of the basic features sets LPCC_SD and MFCC_SD.

Appendix D

Occupation probability of GMM mixtures

Our aim in this section is to show that we can simplify the general case given in [4, chapter 8.7] for the occupation probability of a mixture m in a GMM, leading to the expression

$$\gamma_m(t) = \frac{w_m b_m(\mathbf{o}(t))}{b(\mathbf{o}(t))} \tag{D.1}$$

Generally, for a single stream HMM the occupation probability for state j and mixture m at time t is given by (see [4, chapter 8.7])

$$\gamma_{jm}(t) = \frac{1}{P(\mathbf{O}|\lambda)} U_j(t) w_{jm} b_{jm}(\mathbf{o}(t))\, \beta_j(t) \tag{D.2}$$

where

$$U_j(t) = \begin{cases} \pi_j & \text{if } t = 1 \\ \sum_{i=1}^{N} \alpha_i(t-1) a_{ij} & \text{otherwise} \end{cases} \tag{D.3}$$

We have rewritten these formulae for a single observation sequence $\mathbf{O}$ and discarded thus the index r in the original notation.

Since a GMM owns only a single emitting state (denoted by the index 1) we find for the forward probabilities:

$$\alpha_1(1) = \pi_1 b_1(\mathbf{o}(1)) \tag{D.4}$$

$$\alpha_1(t) = \pi_1 a_{11}^{(t-1)} \prod_{\tau=1}^{t} b_1(\mathbf{o}(\tau)) \text{ for } 1 < t \leq T \tag{D.5}$$

and in particular

$$\alpha_1(t-1) = \frac{\alpha_1(t)}{a_{11} b_1(\mathbf{o}(t))}. \tag{D.6}$$

We can merge both cases of D.3 and write

$$U_1(t) = \pi_1 a_{11}^{(t-1)} \prod_{\tau=1}^{t-1} b_1(\mathbf{o}(\tau)) \text{ for } 1 \leq t \leq T. \tag{D.7}$$

Similar to the forward probabilities we find for the backward probabilities:

$$\beta_1(T) = (1 - a_{11}) \tag{D.8}$$

$$\beta_1(t) = a_{11}^{(T-t)}(1 - a_{11}) \prod_{\tau=t+1}^{T} b_1(\mathbf{o}(\tau)) \text{ for } 1 \leq t < T \tag{D.9}$$

The posterior probability can be expressed easily by the state transition probabilities and the emission probabilities due to the fact, that only a single state sequence is possible:

$$P(\mathbf{O}|\lambda) = \pi_1 a_{11}^{(T-1)}(1 - a_{11}) \prod_{\tau=1}^{T} b_1(\mathbf{o}(\tau)) \tag{D.10}$$

Finally this gives:

$$\gamma_{1m}(t) = \frac{\alpha_1(t) w_{1m} b_{1m}(\mathbf{o}(t))\, \beta_1(t)}{b_1(\mathbf{o}(t))\, \pi_1 a_{11}^{(T-1)}(1 - a_{11}) \prod_{\tau=1}^{T} b_1(\mathbf{o}(\tau))} \tag{D.11}$$

$$= \frac{w_{1m} b_{1m}(\mathbf{o}(t))}{b_1(\mathbf{o}(t))} \tag{D.12}$$

Since we have only a single emitting state in a GMM, we can leave out the state indices $j = 1$:

$$\gamma_m(t) = \frac{w_m b_m(\mathbf{o}(t))}{b(\mathbf{o}(t))} \tag{D.13}$$

Appendix E

"B-" and "BW-" training and evaluation sets

		Item set							
		B-Base tr	B-Base ev	B-FQ tr	B-FQ ev	B-F tr	B-Q tr	B-G tr	B-N tr
No. of items		20	22	20	22	22	22	22	22
Session type	**Session**								
	01	P1-P6		P1-P6		P1-P6	P1-P6		
	03	P1		P1-P5,P6		P1-P5,P7	P1-P4		
	05			P1-P4,P6,P7		P1,P2			
FQ	09		P3,P5	P1,P2	P3,P5-P7				
	12		P4		P1,P2,P4-P7				
	13		P1,P7		P1,P3-P7				
	17		P6,P7		P2-P7				
	07	P1-P3				P1-P6			P1-P7
FN	14		P4,P7						P1-P3
	19		P4,P7						
	02	P1-P6					P1-P6	P1-P6	
	06	P1					P1-P4	P1-P5,P7	
	08							P1,P2	
GQ	11		P3,P5						
	15		P4						
	18		P1,P7						
	20		P6,P7						
	04	P1-P3						P1-P6	P1-P7
GN	10		P4,P7						P1-P3
	16		P4,P7						

Table E.1: Training sets (suffix tr) and evaluation sets (suffix ev) of the "B-" series. The evaluation sets do not contain any of the items of the training sets.

		Item set				
		BW-Base tr	BW-F tr	BW-Q tr	BW-G tr	BW-N tr
No. of items		42	42	42	42	42
Session type	**Session**					
	01	P1-P6	P1-P6	P1-P6		
	03	P1	P1-P5,P7	P1-P4		
	05		P1,P2			
FQ	09	P3,P5	P3,P5,P7	P3,P5,P7		
	12	P4	P1,P4,P5,P7	P4,P5		
	13	P1,P7	P1,P4,P6,P7	P1,P4,P7		
	17	P6,P7	P3,P4,P6,P7	P3,P6,P7		
	07	P1-P3	P1-P6			P1-P7
FN	14	P4,P7	P4,P5,P7			P1-P7
	19	P4,P7	P2,P4,P6,P7			P1-P7
	02	P1-P6		P1-P6	P1-P6	
	06	P1		P1-P4	P1-P5,P7	
	08				P1,P2	
GQ	11	P3,P5		P3,P5,P7	P3,P5,P7	
	15	P4		P4,P5	P1,P4,P5,P7	
	18	P1,P7		P1,P4,P7	P1,P4,P6,P7	
	20	P6,P7		P3,P6,P7	P3,P4,P6,P7	
	04	P1-P3			P1-P6	P1-P7
GN	10	P4,P7			P4,P5,P7	P1-P7
	16	P4,P7	P4,P7		P2,P4,P6,P7	P1-P7

Table E.2: World training sets of the "BW-" series. Each set contains 42 items.

Appendix F

Sub-corpus with manual phonetic segmentation

	Q sessions		N sessions	
speaker	**FQ**	**GQ**	**FN**	**GN**
0001			14	
0011	12			
0019		02		
0022			07	
0027			19	
0028			07	
0039			14	
0044	03			
0047		08		
0048	12	11	14	04
0049	12	20	07	16
0059	13			
0063	13			
0068	13			
0070		08		
0075	12			
0077		15		
0087	13			
0097	13			
0105		20		
0113	05			
0116				15
0119	13			
0128		08		
0131			07	
0135		20		
0136	17			
0142	01			
0144	05			
0149			14	
Total #	13	7	8	2

Table F.1: Session type and session number of manually segmented utterances (F1 items). Speakers 0048 and 0049 provide one session of each environment and network type, while from the remaining speaker only a single session was randomly drawn.

Appendix G

Adaptation results

All results reported here are based on the simulated GSM adaptation data with clean transmission channel, recording set 'sim G FR' (see the description of the recording sets in section 7.3). Only two additional experiments are done for the 32-mixture GMMs using the adaptation data from simulated GSM data with distorted channel, namely recording sets 'sim G FR C-0.08' and 'sim G FR C-0.13'. These results are given in table G.15 and table G.16.

G.1 Data split S2, number triplets (items P1 - P7)

G.1.1 HMM with subwords, 1 mixture

Adaptation with simulated GSM, test with simulated GSM (adaptation scenario 'sim-sim')

Common properties

Training world model:	S2 Q
Training client model:	S2 FQ training
Adaptation client model:	S2 simGQ training
Evaluation:	S2 simGQ evaluate

Performance in matched and mismatched case

Training data	Evaluation data	EER
S2 FQ training	S2 FQ evaluate	4.1
S2 FQ training	S2 simGQ evaluate	9.1
S2 simGQ training	S2 simGQ evaluate	5.6

Table G.1: General properties and performance with matched and mismatched data.

MLLR type	**Clustering type**				
	Single	**Ac2**	**Ac6**	**Ph2**	**Ph6**
Bias	7.2	6.6	6.0	7.2	7.1
Bias + VarScale	5.8	5.6	5.5	5.9	6.7
Full	6.3	6.5	6.6	6.4	6.7
Full + VarScale	5.5	5.7	5.8	5.9	6.1

Table G.2: Performance using 'per client' transformation sets.

MLLR type	**Clustering type**				
	Single	**Ac2**	**Ac6**	**Ph2**	**Ph6**
Bias	7.7	7.7	7.6	7.7	7.8
Bias + VarScale	7.9	7.9	7.8	7.9	8.1
Full	8.5	9.8	14.4	9.8	14.3
Full + VarScale	8.5	9.6	14.1	9.9	13.9

Table G.3: Performance using 'common' transformation set.

Adaptation with simulated GSM, real GSM for test (adaptation scenario 'sim-real')

Common properties	
Training world model:	S2 Q
Training client model:	S2 FQ training
Adaptation client model:	S2 simGQ training
Evaluation:	S2 GQ evaluate

Performance in matched and mismatched case		
Training data	**Evaluation data**	**EER**
S2 FQ training	S2 FQ evaluate	4.1
S2 FQ training	S2 GQ evaluate	19.9
S2 GQ training	S2 GQ evaluate	7.5

Table G.4: General properties and performance with matched and mismatched data.

MLLR type	**Clustering type**				
	Single	**Ac2**	**Ac6**	**Ph2**	**Ph6**
Bias	18.7	18.7	18.4	18.6	18.5
Bias + VarScale	17.4	16.4	16.7	17.2	16.6
Full	18.3	18.0	18.3	18.4	18.4
Full + VarScale	17.1	17.3	17.2	17.4	17.7

Table G.5: Performance using 'per client' transformation sets.

MLLR type	**Clustering type**				
	Single	**Ac2**	**Ac6**	**Ph2**	**Ph6**
Bias	19.1	19.0	18.8	19.2	19.3
Bias + VarScale	18.9	18.5	18.8	19.0	18.8
Full	20.4	21.1	25.7	21.6	26.0
Full + VarScale	20.3	20.8	26.3	21.6	25.6

Table G.6: Performance using 'common' transformation set.

Adaptation with real GSM, real GSM for test (adaptation scenario 'real-real')

Common properties	
Training world model:	S2 Q
Training client model:	S2 FQ training
Adaptation client model:	S2 GQ training
Evaluation:	S2 GQ evaluate

Performance in matched and mismatched case

Training data	Evaluation data	EER
S2 FQ training	S2 FQ evaluate	4.1
S2 FQ training	S2 GQ evaluate	19.9
S2 GQ training	S2 GQ evaluate	7.5

Table G.7: General properties and performance with matched and mismatched data.

MLLR type	Clustering type				
	Single	Ac2	Ac6	Ph2	Ph6
Bias	9.2	8.0	7.5	9.8	8.6
Bias + VarScale	6.7	6.5	6.3	7.4	6.7
Full	8.1	7.7	8.4	8.1	8.4
Full + VarScale	7.1	6.9	7.9	7.4	7.4

Table G.8: Performance using 'per client' transformation sets.

MLLR type	Clustering type				
	Single	Ac2	Ac6	Ph2	Ph6
Bias	18.9	18.5	18.7	18.8	18.9
Bias + VarScale	18.7	18.0	18.4	18.7	18.6
Full	22.0	22.2	30.2	26.3	35.5
Full + VarScale	21.9	21.6	30.5	26.5	35.6

Table G.9: Performance using 'common' transformation set.

G.1.2 GMMs, 32 mixtures

Adaptation with simulated GSM, test with simulated GSM (adaptation scenario 'sim-sim')

Common properties	
Training world model:	S2 Q
Training client model:	S2 FQ training
Adaptation client model:	S2 simGQ training
Evaluation:	S2 simGQ evaluate

Performance in matched and mismatched case		
Training data	**Evaluation data**	**EER**
S2 FQ training	S2 FQ evaluate	5.4
S2 FQ training	S2 simGQ evaluate	10.4
S2 simGQ training	S2 simGQ evaluate	7.3

Table G.10: General properties and performance with matched and mismatched data.

MLLR type	**Clustering type**				
	Single	**Ac2**	**Ac6**	**Ph2**	**Ph6**
Bias	8.5	8.8	8.5	8.6	8.8
Bias + VarScale	7.6	7.4	7.5	8.1	7.7
Full	7.9	7.9	7.9	7.9	7.9
Full + VarScale	7.3	7.2	7.1	7.3	7.1

Table G.11: Performance using 'per client' transformation sets.

MLLR type	**Clustering type**				
	Single	**Ac2**	**Ac6**	**Ph2**	**Ph6**
Bias	9.2	9.4	9.3	9.3	9.3
Bias + VarScale	9.1	9.2	9.3	9.4	9.4
Full	12.8	23.8	22.5	20.0	22.2
Full + VarScale	12.9	22.8	22.0	19.6	22.4

Table G.12: Performance using 'common' transformation set.

Adaptation with simulated GSM, real GSM for test (adaptation scenario 'sim-real')

Common properties	
Training world model:	S2 Q
Training client model:	S2 FQ training
Adaptation client model:	S2 simGQ training
Evaluation:	S2 GQ evaluate

Performance in matched and mismatched case

Training data	Evaluation data	EER
S2 FQ training	S2 FQ evaluate	5.4
S2 FQ training	S2 GQ evaluate	28.9
S2 GQ training	S2 GQ evaluate	10.5

Table G.13: General properties and performance with matched and mismatched data.

MLLR type	Clustering type				
	Single	Ac2	Ac6	Ph2	Ph6
Bias	26.7	27.2	27.1	26.9	27.3
Bias + VarScale	24.7	24.9	24.9	25.2	24.6
Full	25.4	25.1	25.0	25.0	25.0
Full + VarScale	24.3	24.1	23.6	24.2	23.7

Table G.14: Performance using 'per client' transformation sets.

MLLR type	Clustering type				
	Single	Ac2	Ac6	Ph2	Ph6
Bias	26.4	27.0	26.7	26.4	26.7
Bias + VarScale	25.5	25.4	25.8	25.0	25.4
Full	24.7	24.9	24.7	24.8	24.7
Full + VarScale	25.8	24.8	24.4	25.1	24.1

Table G.15: Performance using 'per client' transformation sets and simulated GSM recordings taken from the set 'sim G FR C-0.08'

MLLR type	**Clustering type**				
	Single	**Ac2**	**Ac6**	**Ph2**	**Ph6**
Bias	26.5	27.0	26.8	26.6	26.7
Bias + VarScale	25.3	25.2	25.9	25.0	25.3
Full	25.0	24.9	24.6	24.9	24.8
Full + VarScale	25.5	24.7	24.4	25.0	24.1

Table G.16: Performance using 'per client' transformation sets and simulated GSM recordings taken from the set 'sim G FR C-0.13'

MLLR type	**Clustering type**				
	Single	**Ac2**	**Ac6**	**Ph2**	**Ph6**
Bias	27.0	26.8	26.6	27.1	26.7
Bias + VarScale	25.8	25.9	25.8	25.8	26.0
Full	29.5	38.4	37.3	36.3	36.5
Full + VarScale	28.6	36.3	36.5	35.8	37.2

Table G.17: Performance using 'common' transformation set.

Adaptation with real GSM, real GSM for test (adaptation scenario 'real-real')

Common properties	
Training world model:	S2 Q
Training client model:	S2 FQ training
Adaptation client model:	S2 GQ training
Evaluation:	S2 GQ evaluate

Performance in matched and mismatched case

Training data	Evaluation data	EER
S2 FQ training	S2 FQ evaluate	5.4
S2 FQ training	S2 GQ evaluate	28.9
S2 GQ training	S2 GQ evaluate	10.5

Table G.18: General properties and performance with matched and mismatched data.

MLLR type	Clustering type				
	Single	Ac2	Ac6	Ph2	Ph6
Bias	12.5	11.8	10.8	12.2	11.1
Bias + VarScale	11.2	9.6	9.3	10.5	9.5
Full	9.7	9.8	9.8	9.9	9.8
Full + VarScale	9.3	9.3	9.4	9.2	9.1

Table G.19: Performance using 'per client' transformation sets.

MLLR type	Clustering type				
	Single	Ac2	Ac6	Ph2	Ph6
Bias	26.0	26.2	26.1	25.9	26.2
Bias + VarScale	25.1	25.4	26.0	25.7	26.0
Full	28.8	37.3	37.1	36.3	35.9
Full + VarScale	28.1	36.2	36.0	36.1	36.2

Table G.20: Performance using 'common' transformation set.

G.2 Items F1, GMMs 32 mixtures

Adaptation with simulated GSM, test with simulated GSM (adaptation scenario 'sim-sim')

Common properties	
Training world model:	F1 Q
Training client model:	F1 FQ training
Adaptation client model:	F1 simGQ training
Evaluation:	F1 simGQ evaluate

Performance in matched and mismatched case		
Training data	**Evaluation data**	**EER**
F1 FQ training	F1 FQ evaluate	2.0
F1 FQ training	F1 simGQ evaluate	13.1
F1 simGQ training	F1 simGQ evaluate	1.8

Table G.21: General properties and performance with matched and mismatched data.

MLLR type	**Clustering type**				
	Single	**Ac2**	**Ac6**	**Ph2**	**Ph6**
Bias	4.9	4.6	4.0	4.3	–
Bias + VarScale	–	–	–	–	–
Full	2.2	2.1	2.1	2.1	–
Full + VarScale	2.0	1.9	1.9	1.9	–

Table G.22: Performance using 'per client' transformation sets.

MLLR type	**Clustering type**				
	Single	**Ac2**	**Ac6**	**Ph2**	**Ph6**
Bias	–	–	–	–	–
Bias + VarScale	–	–	–	–	–
Full	–	–	–	–	–
Full + VarScale	5.7	–	5.8	–	–

Table G.23: Performance using 'common' transformation set.

Appendix H

Formulary and abbreviations

$P(X)$	the probability of an event X
$p(x)$	the probability density function for a continuous value x
$P(X\|Y)$	the conditional probability for an event X under the condition that Y occurred
N	number of states in a HMM
M	number of mixtures components in a given state
T	number of observations
n	dimensionality of the observation vector
$\mathbf{O}$	a sequence of observation vectors
$\mathbf{o}$	a single observation vector
$\mathbf{o}(t)$	the observation vector at time t, $1 \leq t \leq T$
a_{ij}	the probability of a transition from state i to state j
π_j	initial probability for starting a state sequence in state j
$b_j(\mathbf{o}(t))$	probability density function in state j for observations
$b_{jm}(\mathbf{o}(t))$	probability density function of component m in state j for observations
w_{jm}	the mixture weight of component m of state j
$\boldsymbol{\mu}_{jm}$	the vector of means for the mixture component m of state j
$\boldsymbol{\Sigma}_{jm}$	the covariance matrix for mixture component m of state j
λ	the set of all parameters defining a HMM
$\gamma^r_{jm}(t)$	probability of occupying the mixture component m of state j at time t for the r'th observation sequence
$\gamma_{jm}(t)$	probability of occupying the mixture component m of state j at time t for the current observation sequence
$\gamma_j(t)$	probability of occupying state j at time t for the current observation sequence

Acknowledgements

First, I would like to thank my supervisors, Prof. Günther Ruske and Dr. Florian Schiel for their constant support, comments and questions during the dissertation project. I also appreciate very much the support of the Bavarian Archive for Speech Signals for funding the final months of this dissertation.

I also thank Prof. Andreas Herkersdorf and Prof. Georg Färber for their interest in the project.

Furthermore I would like to thank, in no particular order:

- Marion Libossek for access to her excellent collection of papers.
- Dr. Mark Gales for discussions and explanations on HTK's adaption source code.
- Klaus Jänsch for answering many question regarding distributed computing.
- Christine Enzinger for proof reading and our regular breaks with cigarettes (she) and tea (me).
- My further proof readers Angie Kempe and Christian Sunkler.
- Stefanie Lukasz for taking on the role as master proof reader and "discoverer of misleading argumentation". Her support was always encouraging.
- Ulrich Reubold for the manual phonetic segmentation of a small part of the VeriDat corpus that was used in section 7.5.
- Prof. Christian Reinsch for some very helpful insights into the numerical aspects of the Cholesky decomposition.
- Dr. Hartmut Pfitzinger for having several short but inspiring discussions.
- Dr. Christoph Draxler for his help especially on the VeriDat database and other questions.
- The members of the HTK mailing lists for getting started with HTK and discussing the usage of its features.
- Karl Weilhammer for the LaTeX template of his dissertation.
- My room mates Hannes Mögele and Krystsina Pröll for the nice work atmosphere (and Krystsina's Kinder-Schokolade she offered from time to time).

Finally I would like to thank my parents for their constant support and encouragement.

Bibliography

[1] L. G. Kersta. Voiceprint Identification. *Nature*, 196:1253–1257, 1962. Cited on page 1.

[2] S. Pruzansky. Pattern-Matching Procedure for Automatic Talker Recognition. *The Journal of the Acoustical Society of America*, 35(6):354–358, 1963. Cited on page 1.

[3] R. Lippmann, E. Martin, and D. Paul. Multi-Style Training for Robust Isolated-Word Speech Recognition. pages 705–708. Proceedings ICASSP 87, 1987. Cited on page 2.

[4] S. Young et al. The HTK Book, Cambridge University, `http://htk.eng.cam.ac.uk`, 2004. Cited on pages 3, 14, 19, and 153.

[5] A.J. Mansfield and J.L. Wayman. *Best Practices in Testing and Reporting Performance of Biometric Devices*. Version 2.01, National Physical Laboratory, England, 2002. Cited on pages 3, 9, 22, 28, 30, 32, 33, 35, 45, and 139.

[6] D. Polemi. Biometric Techniques Review and Evaluation of Biometric Techniques for Identification and Authentification, Including an Appraisal of the Areas Where They Are Most Applicable, Community Research & Development Information Service, European Union, `ftp://ftp.cordis.europa.eu/pub/infosec/docs/biomet.doc`, 1997. Cited on page 5.

[7] A. Jain, R. Bolle, and S. Pankanti (edts.). *Biometrics: Personal Identification in a Networked Society*. Kluwer Academics Publishers, Boston, 1999. Cited on page 5.

[8] G. Doddington, W. Ligget, A. Martin, M. Przybocki, and D. Reynolds. SHEEP, GOATS, LAMBS and WOLVES - A Statistical Analysis of Speaker Performance in the NIST 1998 Speaker Recognition Evaluation. In *Proceedings of ICSLP 98*, pages 1351–1354, 1998. Cited on pages 9, 33, 45, and 139.

[9] J.W. Koolwaaij and L. Boves. A New Procedure for Classifying Speakers in Speaker Verification Systems. In *Proceedings Eurospeech*, pages 2355–2358, 1997. Cited on page 10.

[10] S.H. Maes. Conversational Biometrics. In *Proceedings Eurospeech 99*, 1999. Cited on page 12.

[11] Sadaoki Furui. Speaker-dependent-feature extraction, recognition and processing techniques. *Speech Communication*, (10):505–520, 1991. Cited on page 13.

[12] S. Furui. *An Overview of Speaker Recgnition Technology*. Kluwer Academic Publishers, 1996. Cited on page 13.

[13] G.R. Doddington. Speaker Recognition - Identifying People By Their Voices. *Proceedings of the IEEE*, 73:1651–1664, 1985. Cited on page 13.

[14] F. Bimbot, H.-P. Hutter, C. Jaboulet, J. Koolwaaij, J. Lindberg, and J.-B. Pierrot. Speaker verification in the telephone network: Research activities in the CAVE project. In *Proceedings Eurospeech*, 1997. Cited on page 13.

[15] J. P. Campbell. Speaker Recognition - A Tutorial. *Proceedings of the IEEE*, 85(9):1437–1462, 1997. Cited on page 13.

[16] R.W. Schafer and L.R. Rabiner. *Digital Representations of Speech Signals*. Morgan Kaufmann Publishers, 1975. Cited on page 14.

[17] T.F. Quatieri. *Discrete-Time Speech Processing: Principles and Practice*. Prentice-Hall, 2001. Cited on page 14.

[18] Furui. Cepstral analysis technique for automatic speaker verification. *IEEE Transactions on Acustic, Speech, Signal Processing*, ASSP-29(2):254–272, 1981. Cited on pages 14 and 18.

[19] S.B. Davis and P. Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continously Spoken Sentences. *IEEE Trans. Acoust., Speech, Signal Processing*, ASSP-28:357–366, 1980. Cited on page 14.

[20] J. D. Markel and A.H. Gray. *Linear Prediction of Speech*. Springer-Verlag, Berlin, Germany, 1976. Cited on pages 15 and 74.

[21] B. S. Atal. Effectiveness of Linear Prediction Characteristics of the Speech Wave for Automatic Speaker Identification and Verification. *JASA*, 55(6):1304–1312, 1974. Cited on page 15.

[22] L.R. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, NJ, 1993. Cited on pages 15 and 16.

[23] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood From Incomplete Data Via the EM Algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977. Cited on pages 16 and 100.

[24] T. K. Moon. The Expectation-Maximization Algorithm. *IEEE Signal Processing Magazin*, 13(6):47–60, 1996. Cited on page 16.

[25] L.R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286, 1989. Cited on page 16.

[26] R.M. Gray and A. Gersho. *Vector Quantization and Signal Compression*. Klewer, Boston, MA, 1991. Cited on page 16.

[27] D. Reynolds. Speaker identification and verification using gaussian mixture speaker models. In *Proceedings Esca Workshop on Speaker Recognition, Identification and Verification*, 1994. Cited on page 17.

[28] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2001. Cited on pages 17 and 105.

[29] F. Bimbot and G. Chollet. *Assessment of Speaker Verification Systems*, pages 408–480. Mouton de Gruyter, 1997. Cited on pages 17, 22, 24, and 28.

[30] A. Rosenberg, J. DeLong, B-H. Juang, and F.Soong. The Use of Cohort Normalized Scores for Speaker Verification. In *Proceedings ICASSP*, volume 2, pages 599–602, 1992. Cited on page 18.

[31] A.E. Rosenberg and S. Parthsarathy. Speaker Background Models for Connected Digit Password Speaker Verification. In *Proceedings ICASSP 96*, 1996. Cited on page 18.

[32] T. Isobe and J. Takahashi. Text-Independent Speaker Verification Using Virtual Speaker Based Cohort Normalization. In *Proceedings Eurospeech*, 1999. Cited on page 18.

[33] A.M. Ariyaeeinia and P. Sivakumaran. Analysis and Comparison of Score Normalisation Methods for Text-Dependent Speaker Verification. pages 1379–1382. Eurospeech 97, 1997. Cited on page 18.

[34] K.-P. Li and J.E. Porter. Normalizations and Selection of Speech Segments for Speaker Recognition Scoring. pages 595–597. Proceedings ICASSP 88, 1988. Cited on page 18.

[35] D.A. Reynolds. Comparison of Background Normalization Methods for Text-Independent Speaker Verification. In *Proceedings Eurospeech*, volume 2, pages 963–966, 1997. Cited on pages 19, 96, and 97.

[36] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas. Score Normalization for Text-Independent Speaker Verification Systems. *Digital Signal Processing*, 10:42–54, 2000. Cited on pages 19 and 96.

[37] U. Türk. MASV website, `http://www.bas.uni-muenchen.de/Bas/SV/`, 2004. Cited on pages 19 and 20.

[38] W. Shen, M. Surette, and R. Khanna. Evaluation of Automated Biometrics-Based Identification and Verification Systems. *Proceedings IEEE*, 85:1464–1479, 1997. Cited on page 22.

[39] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki. The DET Curve in Assessment of Detection Task Performance. In *Proceedings Eurospeech*, pages 1895–1898. Rhodes, Greece, 1997. Cited on pages 27 and 146.

[40] J.L. Wayman. *Confidence Interval and Test Size Estimation for Biometric Data*, pages 91–101. San Jose State University, 2000. Cited on pages 28, 31, 32, 35, 45, and 139.

[41] Günter Clauß, Falk-Rüdiger Finze, and Lothar Partzsch. *Statistik für Soziologen, Pädagogen, Psychologen und Mediziner. Bd. 1: Grundlagen.* Harri Deutsch, 1995. Cited on page 29.

[42] B. D. Jovanovic and P. S. Levy. A Look at the Rule of Three. *The American Statistician*, 51(2):137–139, 1997. Cited on page 30.

[43] G. R. Doddington, M. A. Przybocki, and A. F. Martin. The NIST Speaker Recognition Evaluation - Overview, Methodology, Systems, Results, Perspective. *Speech Communication*, 31:225–254, 2000. Cited on page 30.

[44] G. W. Snedecor and W.G. Cochran. *Statistical Methods*. Iowa State University Press, 8th edition, 1989. Cited on pages 32, 45, and 47.

[45] J. Wayman. *Technical Testing and Evaluation of Biometrical Identification Devices*, pages 345–368. Jain et. al (eds.); Boston, Kluwer Academic Publishers, 1998. Cited on page 33.

[46] M. Schuckers. Some Statistical Aspects of Biometrical Identification Device Performance. *Stats Magazine*, 2001. Cited on pages 33 and 139.

[47] M. Schuckers. Using the Beta-Binomial Distribution to Assess the Performance of a Biometric Identification Device. *International Journal of Image and Graphics*, 2001. Cited on pages 33 and 139.

[48] B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hill, 1993. Cited on page 34.

[49] D. N. Politis. Computer-Intensive Methods in Statistical Analysis. *IEEE Signal Processing Magazine*, 15(1):39–55, 1998. Cited on page 34.

[50] A.M. Zoubir and B. Boashash. The Bootstrap and its Application in Signal Processing. *IEEE Signal Processing Magazine*, 15(1):56–76, 1998. Cited on page 34.

[51] R. M Bolle, N. K. Ratha, and S. Pankanti. Evaluation Authentification Systems Using Bootstrap Confidence Intervals. In *Proceedings of the Empirical Evaluation Methods in Computer Vision*, 2001. Cited on page 35.

[52] S. A. Macskassy, F. J. Provost, and M. L. Littman. Confidence Bands for ROC Curves. *CeDER Working Papers, IS-03-04*, 2003. Cited on page 36.

[53] U. Türk and F. Schiel. Speaker Verification Based on the German VeriDat Database. In *Proceedings Eurospeech*. Geneva, Switzerland, 2003. Cited on pages 41, 64, and 65.

[54] R. D. Zilca. Text-Independent Speaker Verification Using Utterance Level Scoring and Covariance Modelling. *IEEE Trans. Speech and Audio Processing*, 10(6):363–370, 2002. Cited on page 59.

[55] The NIST Speaker Recognition Evaluations, National Institute of Standards and Technology, `http://www.nist.gov/speech/tests/spk/`, 2006. Cited on pages 59 and 146.

[56] M. Kuitert and L. Boves. Speaker Verification With GSM Coded Telephone Speech. In *Proceedings Eurospeech*, 1997. Cited on pages 63, 136, 137, and 140.

[57] F. Bimbot, M .Blomberg, L. Boves, D. Genoud, H.-P. Huttel, C. Jaboulet, J. Koolwaaij, J. Lindberg, and J.-B. Pierrot. An Overview of the CAVE Project Research Activities in Speaker Verification. *Speech Communication*, 31:155–180, 2000. Cited on pages 63 and 140.

[58] ETSI Document EN 300 961. *Digital Cellular Telecommunications System (Phase 2+); Full Rate Speech; Transcoding (GSM 06.10 Version 8.1.1 Release 1999)*. ETSI, 1999. Cited on pages 68 and 71.

[59] J. Degener and C. Bormann. Reference Implementation for GSM RPE-LPT Speech Codec, `ftp://ftp.cs.tu-berlin.de/pub/local/kbs/tubmik/gsm/gsm-1.0.10.tar.gz`, 1994. Cited on page 68.

[60] Open H323, `http://www.openh323.org`. Cited on page 68.

[61] *ITU-T Software Tool Library Manual, Edition 1.0 for the 2000 release of the ITU-T Software Tool Library, Distribution 3.0.* ITU, Genève, Switzerland, 2001. Cited on pages 69 and 70.

[62] T. Salonidis and V. Digalakis. Robust Speech Recognition for Multiple Topological Scenarios of the GSM Mobile Phone System. In *Proceedings ICASSP*, 1998. Cited on page 73.

[63] S. Euler and J. Zinke. The Influence of Speech Coding Algorithms on Automatic Speech Recognition. In *Proceedings ICASSP*, 1994. Cited on page 73.

[64] R. J. Mammone, X. Zhang, and R. P. Ramachandran. Robust Speaker Recognition. *IEEE Signal Processing Magazine*, pages 58–71, 1996. Cited on page 73.

[65] X. Zhang and R.J. Mammone. Channel and Noise Normalization Using Affine Transformed Ceptrsum. In *Proceedings ICSLP*, pages 1993–1996, 1996. Cited on pages 73 and 74.

[66] M. R. Schroeder. Direct (Nonrecursive) Relations Between Cepstrum and Predictor Coefficients. *IEEE Trans. Acoust, Speech and Signal Processing*, 29(2):297–301, 1981. Cited on page 74.

[67] T. Ellbogen. Conventions for Segmentation - BITS Report TP 8/5e, Bavarian Archive for Speech Signals (BAS), `http://www.phonetik.uni-muenchen.de/Forschung/BITS/Conventions_for_segmentation.pdf`, 2004. Cited on pages 75 and 76.

[68] H. Niemann. *Klassifikation von Mustern*. Springer Verlag, 1983. Cited on page 78.

[69] T. Kailath. The Divergence and Bhattacharyya distance measures in Signal Selection. *IEEE Trans. Commun. Tech.*, 15:52–60, 1967. Cited on page 78.

[70] D. Comaniciu and P. Meer. Distribution free decomposition of Multivariate Data. *Pattern Analysis Applications*, 2:22–30, 1999. Cited on page 78.

[71] A. Sankar and C. H. Lee. A Maximum-Likelihood Approach to Stochastic Matching for Robust Speech Recognition. *Transactions on Speech and Audio Processing*, 4(3):190–202, 1996. Cited on pages 83, 84, 86, 88, 93, 94, 95, 98, 100, and 141.

[72] K.-K. Yiu, M.-W. Mak, and S.-Y. Kung. Environment Adaptation for Robust Speaker Verification. In *Proceedings Eurospeech*, 2003. Cited on pages 83, 96, and 97.

[73] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn. Speaker Verification Using Adapted Gaussian Mixture Models. *Digital Signal Processing*, 10:19–41, 2000. Cited on pages 83 and 84.

[74] C. J. Leggetter and P.C. Woodland. Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models. *Computer Speech and Language*, (9):171–185, 1995. Cited on pages 84, 94, 100, and 101.

[75] V. Digalakis, D. Rtischev, and L. Neumeyer. Speaker Adaptation Using Constrained Estimation of Gaussian Mixtures. *Transactions on Speech and Audio Processing*, 3(5):357–366, 1995. Cited on pages 84, 85, and 100.

[76] C. H. Lee, C. H. Lin, and B.H. Juang. A Study on Speaker Adaptation of the Parameters of Continuous Density Hidden Markov Models. *ASSP-39*, 39(4):806–814, 1991. Cited on page 84.

[77] J.-L. Gauvain and C.-H. Lee. Maximum a Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains. *IEEE Transactions on Speech and Audio Processing*, 2(2), 1994. Cited on page 84.

[78] V.J. Leggetter and P.C. Woodland. Speaker Adaptation Using Linear Regression. Technical report, Cambridge University, Engineering Dep., 1994. Cited on pages 84, 87, and 94.

[79] L. Neumeyer, A. Sankar, and V. Digalakis. A Comparative Study of Speaker Adaptation Techniques. In *Proceedings Eurospeech*, pages 1127–1130, 1995. Cited on page 85.

[80] A. Sankar and C.H. Lee. Stochastic Matching for Robust Speech Recognition. *IEEE Signal Processing Letters*, 1:124–125, 1994. Cited on page 86.

[81] E. W.M. Yu, M.-W. Mak, and S.-Y. Kung. Speaker Verification from Coded Telephone Speech Using Stochastic Feature Transformation and Handset Identification. In *Proceedings Pacific-Rim Conference on Multimedia*, pages 598–606, 2002. Cited on pages 86, 94, 95, 96, 97, 135, and 136.

[82] C.-L. Tsang, M.-W. Mak, and S.-Y. Kung. Cluster-Dependent Feature Transformation for Telephone-Based Speaker Verification. In *Proceedings International Conference on Audio- and Video-Based Biometric Person Authentification*, pages 86–94, 2003. Cited on pages 86, 94, 96, 97, 135, and 136.

[83] M.-W. Mak and S.-Y. Kung. Combining Stochastic Feature Transformation and Handset Identification for Telephone-Based Speaker Verification. In *Proceedings ICASSP*, pages 1701–1704, 2002. Cited on pages 86, 94, 95, 96, 97, 135, and 136.

[84] D. A. Reynolds et al. The Effects of Telephone Transmission Degradations on Speaker Recognition Performance. In *Proceedings ICASSP*, pages 329–332, 1995. Cited on page 87.

[85] A. Acero. *Acoustical and Environmental Robustness in Automatic Speech Recognition*. Kluwer Academic Publications, 1992. Cited on page 88.

[86] K.-K. Yiu, M.-W. Mak, M.-C. Cheung, and S.-Y. Kung. A New Approach to Channel Robust Speaker Verification via Constrained Stochastic Feature Transformation. In *Proceedings ICSLP*, 2004. Cited on pages 88, 94, 97, 135, 136, and 141.

[87] K.-K. Yiu, M.-W. Mak, M.-C. Cheung, and S.-Y. Kung. Blind Stochastic Feature Transformation for Speaker Verification over Cellular Networks. In *Proceedings ISIMP*, 2004. Cited on pages 88, 94, 97, 135, 136, and 141.

[88] K. Pröll. Automatische Spracherkennung über das Telefonnetz. Master's thesis, 2004. Cited on page 92.

[89] Various authors. SpeechDat: Public Specifications, `http://www.speechdat.org`, 2006. Cited on pages 93 and 143.

[90] S. S. Kajarekar, L. Ferrer, E. Shriberg, K. Sonmez, A. Stolcke, A. Venkataraman, and J. Zheng. SRI's 2004 NIST Speaker Recognition Evaluation System. volume 1, pages 173–176. Proceedings ICASSP 05, 2005. Cited on pages 93 and 98.

[91] A. Stolcke, L. Ferrer, S. Kajarekar, E. Shriberg, and A. Venkataraman. MLLR Transforms as Features in Speaker Recognition. pages 2425–2429. Proceedings Eurospeech 05, 2005. Cited on pages 93 and 98.

[92] N. Mirghafori, A.O. Hatch, S. Stafford, K. Boakye, D. Gillick, and B. Peskin. ICIS's 2005 Speaker Recognition System. Proceedings IEEE ASRU 2005, 2005. Cited on pages 93 and 98.

[93] D.A. Reynolds. Channel Robust Speaker Verification via Feature Mapping. pages II53–II56. Proceedings ICASSP 03, 2003. Cited on pages 93, 98, 99, 135, 136, and 141.

[94] R. Teunen, B. Shahshahani, and L. Heck. A Model-Based Transformational Approach to Robust Speaker Recognition. volume 2, pages 495–498. Proceedings ICSLP 2000, 2000. Cited on pages 93, 98, and 135.

[95] K.K Yu, M.W. Mak, and S.Y Kung. A GMM-Based Handset Selector for Channel Mismatch Compensation with Applications to Speaker Identification. In *Proceedings Second IEEE Pacific-Rim Conference on Multimedia 2001*, pages 1132–1137, 2001. Cited on pages 94, 135, and 141.

[96] M.W. Mak, M.-C. Cheung, and S.-Y. Kung. Robust Speaker Verification From GSM-Transcoded Speech Based on Decision Fusion and Feature Transformation. In *Proceedings ICASSP*, 2003. Cited on pages 94, 135, and 136.

[97] D.A. Reynolds. HTIMIT and LLHDB: Speech Corpora for the Study of Handset transducer effects. In *Proceedings ICASSP 97*, volume 2, pages 1535–1538, 1997. Cited on page 94.

[98] S.H.Lin, S.Y. Kung, and L.J. Lin. Face Recognition/Detection by Probabilistic Decision-Based Neural Network. *IEEE Transactions on Neural Networks, Special Issue on Biometric Identification*, 8(1):114–132, 1997. Cited on page 96.

[99] M.J.F. Gales and P.C. Woodland. Mean and Variance Adaptation Within the MLLR Framework. *Computer Speech & Language*, 10(4):249–264, 1996. Cited on pages 100 and 102.

[100] M.J.F. Gales. Maximum Likelihood Linear Transformations for HMM-Based Speech Recognition. *Computer Speech & Language*, 12(2):75–98, 1998. Cited on pages 101 and 102.

[101] M. J. F. Gales. The Generation and Use of Regression Class Trees for MLLR Adaptation. Technical report, Cambridge University, Engineering Department, 1996. Cited on page 104.

[102] M.J.F. Gales and P.C. Woodland. Variance Compensation within the MLLR Framework. Technical report, Cambridge University, Engineering Department, 1996. Cited on page 104.

[103] J.L. Wayman. *National Biometric Test Center: Collected Works 1997-2000*. San Jose State University, 2000. Cited on page 139.

[104] F. Bimbot, J.-F. Bonastre, C. Fredouille, G. Gravier, I. Magrin-Chagnolleau, S. Meignier, T. Merlin, J. Ortega-Garcia, D. Petrovska-Delacretaz, and D. A. Reynolds. . *EURASIP Journal on Applied Signal Processing*, (4):430–451, 2004. Cited on page 142.

[105] Various authors. ESPRIT SAM Project 2589: Speech Input and Output Assessment Methodologies and Standardization, `http://www.icp.inpg.fr/Relator/standsam.html`. Cited on page 143.

[106] Joseph Campbell. Testing with The YOHO CD-ROM Voice Verification Corpus. In *Proceedings ICASSP*, 1995. Cited on page 145.

[107] H. Melin. *Databases for Speaker Recognition: Activities in COST250 Working Group 2*. Europeach Commision DG-XIII, Brussels, 2000. Cited on page 145.

[108] J.P.Campbell and D.A.Reynolds. Corpora for the Evaluation of Speaker Recognition Sysems. In *Proceedings ICASSP*, pages 829–832. Phoenix, USA, 1999. Cited on page 145.

[109] J. Godfrey, D. Graff, and A. Martin. Public Databases for Speaker Recognition and Verification. In *Proceedings ESCA Workshop on Autmatic Speaker Recognition, Identification and Verification*, pages 39–42. Martigny, Switzerland, 1994. Cited on page 145.

[110] S. Furui. Research on individuality features in speech waves and automatic speaker recognition techniques. *Speech Communication*, 5(2):183.197, 1986. Cited on page 145.

[111] R. Cole, M. Noel, and V. Noel. The CSLU Speaker Recognition Corpus. In *Proceedings ICSLP*. Sydney, 1998. Cited on page 146.

Index